Part III Chemistry Course L11: Monte Carlo simulations

Ard Louis

January 31, 2004

Introduction

This half of the part III chemistry course *Computer Simulation Methods in Chemistry and Physics* focuses on Monte Carlo techniques.

It should be read together with the other complementary section, written by Dr. Michiel Sprik, which covers Molecular Dynamics techniques.

These lecture notes extensively use:

Daan Frenkel and Berend Smit, “Understanding Molecular Simulation”, (Academic Press, 2002). (referred to as FS2002)

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A. H. Teller and E. Teller, J. Chem. Phys. **21**, 1087-1092 (1953). (referred to as M1953).

There are, by now, quite a number of books that discuss the use of computer simulations in physics and chemistry. A few that we recommend are listed in the notes by Dr. Sprik.

Contents

Chapter 1

Solving integrals using random numbers

1.1 Introduction

The pressure of the second world war stimulated many important technological breakthroughs in radar, atomic fission, cryptography and rocket flight. A somewhat belated, but no less important, advance was the development of the Monte Carlo (MC) method on computers. Three scientists at the Los Alamos National Laboratory in New Mexico, Nicolas Metropolis, John von Neumann, and Stanislaw Ulam, first used the MC method to study the diffusion of neutrons in fissionable materials. Metropolis coined the word "Monte Carlo" – because of their use of random numbers – later (in 1947), although the idea of using statistical sampling to calculate integrals has been around for much longer. A famous early example is named after the French naturalist Comte de Buffon, who, in 1777, showed how to estimate π by throwing a needle at random onto a set of equally spaced parallel lines. This apparently became something of a 19th century party trick: a number of different investigators tried their hand at "Buffon's needle", cumulating in an attempt by Lazzarini in 1901, who claimed to have obtained a best estimate of $\pi \approx 3.1415929$ – an accuracy of 7 significant digits! – by throwing a needle 3408 times onto a paper sheet[1].

A bewildering array of different MC techniques are now applied to an ever increasing number of problems across science and engineering. In the business world, MC simulations are routinely used to asses risk, setting the value of your insurance premium, or to price complex financial instruments such as derivative securities, determining the value of your stock portfolio.

These lectures will focus on the basic principles behind Monte Carlo, and most applications will be to the calculation of properties of simple atomic and molecular systems.

[1] Lazzarini almost certainly doctored his results. You can easily check this by trying one of the many web-based Java applets that do Buffon's needle. See http://www.sas.upenn.edu/ hongkai/research/mc/mc.html for a nice example

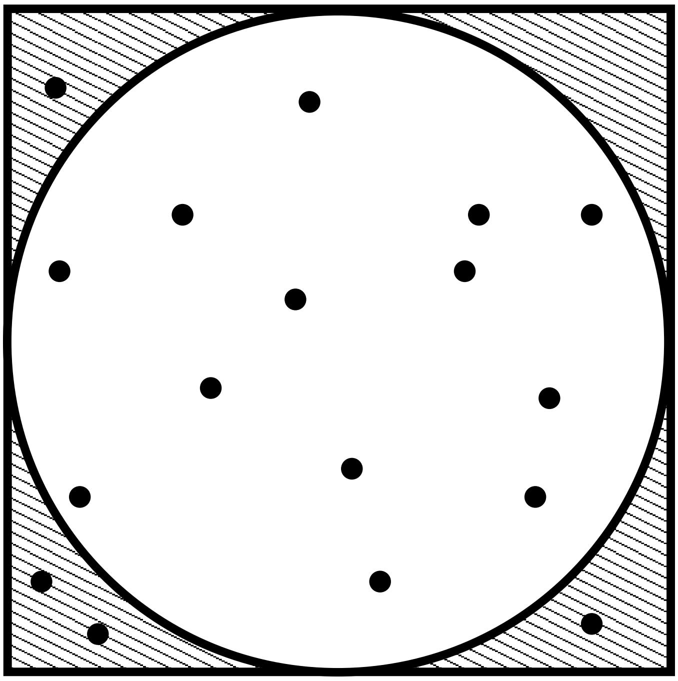


Figure 1.1: π can be calculated by throwing randomly aimed darts at this square, and counting the fraction that lie within the circle. If you do this enough times, this fraction tends toward $\pi/4$. For example, what estimate of π do the random points above give?

1.2 Why stochastic methods work better in high dimensions

But first let us investigate a simple variation of Buffon's party trick: If you were so bad at darts that your throws could be considered as truly random, then it's not hard to see that the probability of having your dart land inside the circle of Fig. 1.1 would be $\pi/4 \approx 0.785$ of the probability of it landing inside the entire square (just compare the areas). So what you are really doing is evaluating a two-dimensional integral (calculating an area) by a stochastic method.

How accurate would this determination of π be? Clearly if you only throw only a few darts, your value can't be very reliable. For example, if you throw three darts, you could find any of the following ratios: $0, \frac{1}{3}, \frac{2}{3}, 1$. The more darts you throw, the better your estimate should become. But just how quickly would you converge to the correct answer? To work this out, it is instructive to simplify even further, and study the stochastic evaluation of 1-dimensional integrals.

An integral, such as the one described in Fig. 1.2, could be evaluated by selecting N random points x_i on the interval $[0, 1]$:

$$I = \int_0^1 dx f(x) \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i) \tag{1.1}$$

A good measure of the error in this average is the standard deviation σ_I, or its square, the variance, given by $\sigma_I^2 \equiv < (I - < I >)^2 > = < I^2 > - < I >^2$, where the brackets $<>$

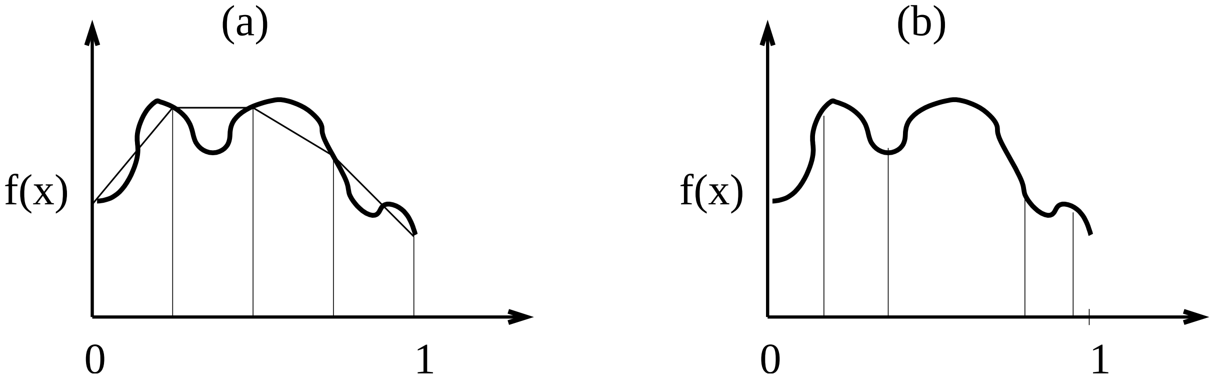


Figure 1.2: This 1-D integral $I = \int_0^1 dx f(x)$ can be calculated in a conventional way (figure (a)), by splitting it up into N segments between 0 and 1, using, e.g. trapezoidal quadrature, or it could be evaluated by MC techniques with random sampling of points (figure (b)). Conventional techniques work best for low dimensions D, whereas MC is better for integrals in high D.

denote an average over many different independent MC evaluations of the integral I. Using Eq. (1.1), the variance can be rewritten as

$$\begin{aligned}
\sigma_I^2 &= \left\langle \left(\frac{1}{N} \sum_{i=1}^{N} f(x_i) - \left\langle \frac{1}{N} \sum_{i=1}^{N} f(x_i) \right\rangle \right)^2 \right\rangle \\
&= \frac{1}{N^2} \left\langle \left(\sum_{i=1}^{N} (f(x_i) - < f(x) >) \right) \left(\sum_{j=1}^{N} (f(x_j) - < f(x) >) \right) \right\rangle \\
&= \frac{1}{N^2} \left\langle \sum_{i=1}^{N} (f(x_i) - < f(x) >)^2 \right\rangle \qquad (1.2)
\end{aligned}$$

where we've used the fact that the $f(x_i)$ are uncorrelated (which is true if the x_i are uncorrelated), first to write $< I >= 1/N \sum_i^N < f(x) >$ (dropping the index i since x_i is a dummy variable), and then again in last line, where the cross-averages between the i and j sums drop out (i.e.$< (f(x_i) - < f >)(f(x_j) - < f >) >= 0$ if $i \neq j$). Eq. (1.2) can rewritten as:

$$\sigma_I^2 = \frac{1}{N} \sigma_f^2 \qquad (1.3)$$

where σ_f^2 is the (average) variance in $f(x)$ itself, i.e., it measures how much $f(x_i)$ deviates from its average value over the integration region[2]. Since σ_f is, to first order, independent of N, the standard deviation, or average error in the MC evaluation of the integral I of Eq. (1.1), scales as $\sigma_I \sim 1/\sqrt{N}$.

In one dimension (D=1) we can do much better with the same amount of effort by using

[2] Even in this derivation there are one or two subtle assumptions that a purist might snipe at. On the other hand, a much easier way to derive this would be to simply invoke the central limit theorem, from which it follows that the total variance σ_{tot}^2 of N independent statistical samples, each with variance σ_i^2 is given by $\sigma_{tot}^2 \approx 1/N < \sigma_i^2 >$.

standard quadrature methods. Even the simple trapezoidal rule:

$$I = \frac{1}{N}\left(\frac{1}{2}f(0) + \sum_{i}^{N-2} f(x_i) + \frac{1}{2}f(1)\right) + \mathcal{O}(\frac{1}{N^3}), \tag{1.4}$$

where the $x_i = i/N$ are equally spaced on $[0,1]$, scales much better than the MC algorithm. For example, if you quadruple the number of points, the error E_{trap} in trapezoidal rule would go down by a factor $1/4^3 = 64$, while the error in the MC evaluation would merely drop by a factor of 2. In fact, the $1/\sqrt{N}$ scaling implies that if you want to increase the accuracy of your MC calculation by one order of magnitude, you need to do 100 times as much work. So MC clearly isn't the most fruitful way to calculate 1-D integrals such as those shown in Fig. 1.2.

The advantages of Monte-Carlo methods only emerge in higher dimensions. This can be seen from simple scaling arguments. Consider an integral over a D dimensional hypercube. Using a standard quadrature method such as the trapezoidal rule, with a fixed spacing of M points per dimension, still gives an error of $E_{trap} \propto M^{-3}$. However now the total number of points is $N = M^D$. The cost of the calculation is proportional to N, which counts the number of independent evaluations of $f(x_i)$ you need to make. Therefore the error in the integral I scales as:

$$E_{trap} \propto M^{-3} = N^{-3/D}. \tag{1.5}$$

In MC, however, the integral's error E_{MC} is independent of dimension (just check the derivation of Eq. (1.2)), and would still scale as

$$E_{MC} \equiv \sigma_I \propto N^{-\frac{1}{2}} \tag{1.6}$$

In other words, MC becomes more efficient than the trapezoidal rule roughly when $N^{-3/D} > N^{-1/2}$, or for dimensions higher than $D = 6$. There are more accurate quadrature methods than the trapezoidal rule, but the errors typically scale with the discretisation M, and so for large enough N, MC will always become more efficient. The reason that MC and related stochastic methods are so popular is because many problems in science and engineering require the evaluation of very high dimensional integrals.

1.3 Importance sampling

Sampling points from a uniform distribution, as done in the previous section, may not be the best way to perform a MC calculation. Consider, for example, Fig. 1.3, where most of the weight of the integral comes from a small range of x where $f(x)$ is large. Sampling more often in this region should increase the accuracy of the MC integration. To make this more concrete, let's assume that we are sampling the points from some (positive definite) normalised probability distribution $w(x)$. The integral in Fig. 1.3 would be rewritten as

$$I = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_{i/w})}{w(x_{i/w})} \tag{1.7}$$

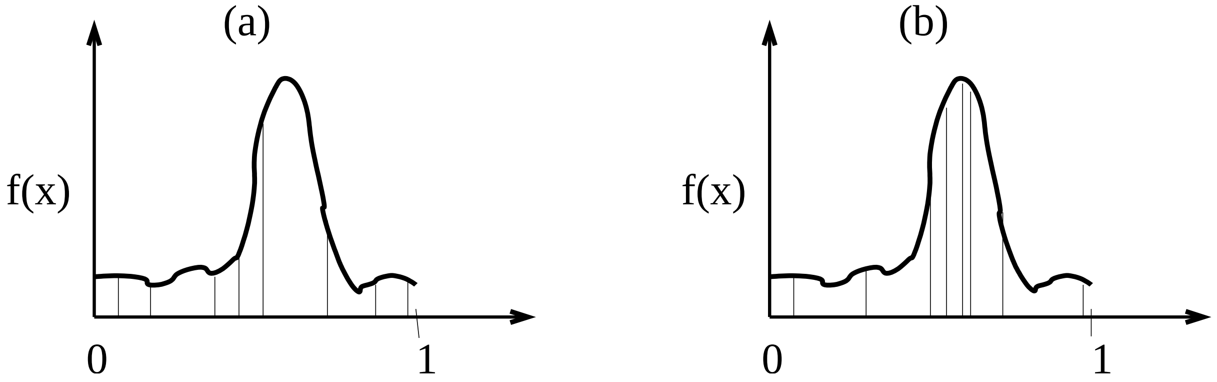


Figure 1.3: In figure (a) the N random points x_i are chosen from a uniform distribution, while figure (b) they come from a biased distribution where points are more likely to occur on a range where $f(x)$ is large. This "importance sampling" can greatly increase the accuracy of a MC integral evaluation.

where the w in $x_{i/w}$ is added to emphasise that the x are drawn from the distribution $w(x)$. The division by $w(x_{i/w})$ compensates for the "biasing" of the distribution of the $x_{i/w}$[3]. (Setting $w(x) = 1$ would reduce to uniform sampling.) A very similar analysis to Eq. (1.2) results in the following expression for the variance:

$$\sigma^2_{I/w} \approx \frac{1}{N}\left\langle \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f(x_{i/w})}{w(x_{i/w})} - < \frac{f(x_{i/w})}{w(x_{i/w})} >\right)^2\right\rangle = \frac{1}{N}\sigma^2_{f/w}. \tag{1.8}$$

Choosing a different sampling distribution $w(x)$ hasn't changed the scaling with N, but it has changed the pre-factor. The best pre-factor would result from using $w(x) = f(x)/ < f >$, in which case $\sigma_{f/w} = 0$! Unfortunately in MC, as in life, there is no such thing as a free lunch: you would need to first know the full integral to obtain $< f >$, which rather defeats the purpose. In practise, however, a good estimate of $w(x)$ may still be available, leading to appreciable gains in accuracy (see the example). Choosing such a distribution is called **importance sampling**, and is a mainstay of almost any MC calculation.

In many cases, especially when sampling highly non-uniform functions, brute force MC evaluations with a uniform distribution of x can result in very large prefactors to the $\sigma_I \propto N^{\frac{1}{2}}$ scaling law[4]. This is exactly the case for the statistical mechanics of atomic and molecular systems, where the high dimensional integrals have a significant contribution in only a very small fraction of the total possible sampling space. Trying to evaluate these integrals without importance sampling would be impossible.

[3] For a more careful derivation see e.g. FS2002, p 25, or try to work it out for yourself by changing variables like you would do for an integration problem.

[4] It is tempting to use the sum in Eq.(1.8), with the $f(x_i)$ drawn from a single run, as an "on the fly" estimator of $\sigma_{f/w}$. This can be dangerous. Consider the example of uniform sampling points in Fig. 1.3(a): $\sigma_{f/w}$ will be significantly underestimated because the large peak that dominates the integral is not sampled well.

Example of 1-d importance sampling

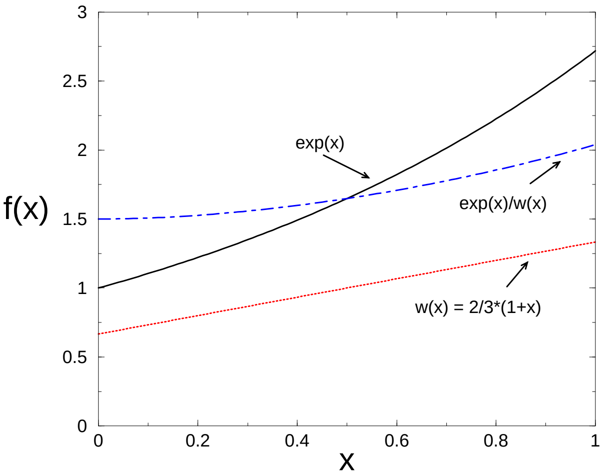
From Eq. (1.2) it follows that the error in an N-step uniform distribution MC evaluation of the integral $I = \int_0^1 dx \exp[x] \approx 1.7182818$ is

$$\sigma_I \approx \frac{0.5}{\sqrt{N}} \tag{1.9}$$

If the normalised importance sampling function $w(x) = 2/3 * (1 + x)$ is used, then the new error, calculated with Eq. (1.8) is:

$$\sigma_{I/w} \approx \frac{0.16}{\sqrt{N}} \tag{1.10}$$

Importance sampling leads to a factor 3 gain in accuracy.



1.4 The Metropolis Monte Carlo Method

1.4.1 Integrals relevant to statistical mechanics

The summit of statistical mechanics

Statistical mechanics tells us that the probability p_i of finding a system at constant number N, volume V and temperature T in a microstate i with total energy E_i is proportional to

$$p_i = \frac{\exp\left[-\beta E_i\right]}{Q(N,V,T)} \tag{1.11}$$

where the inverse temperature $\beta = 1/k_B T$, and k_B is Boltzmann's constant. The partition function $Q(N,V,T)$ is defined as the sum over all states:

$$Q(N,V,T) = \sum_i \exp\left[-\beta E_i\right] \tag{1.12}$$

and the average of an operator A is given by:

$$< A >= \sum_i p_i A_i = \frac{1}{Q} \sum_i \exp\left[-\frac{E_i}{k_B T}\right] A_i \tag{1.13}$$

where A_i is the physical value of A for state i. In the words of the great physicist Richard Feynman:

> "*This fundamental law is the summit of statistical mechanics, and the entire subject is either a slide-down from this summit, as the principle is applied to various cases, or the climb-up to where the fundamental law is derived ...*"[5]

We now will begin with slide-down to computing averages with MC techniques.

Why applications of MC to statistical mechanics must use importance sampling

The simplest way to calculate the average in Eq. (1.13) with MC would be to choose M states at random and average:

$$A_M = \frac{\sum_i^M A_i \exp\left[-\beta E_i\right]}{\sum_i^M \exp\left[-\beta E_i\right]}. \tag{1.14}$$

In the limit that $M \to \infty$, $A_M \to < A >$. There are two major problems with this approach:

1. The number of state points in a statistical mechanical system grows exponentially with system size.

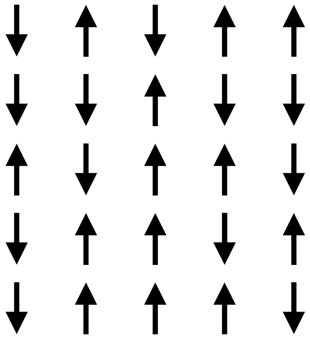
2. Averages like Eq. (1.14) are typically dominated by a small fraction of these states, which random sampling from a uniform distribution will almost certainly miss.

[5] R.P. Feynman, "Statistical Mechanics", Addison-Wesley, 1972, p1

Consider the following two examples:

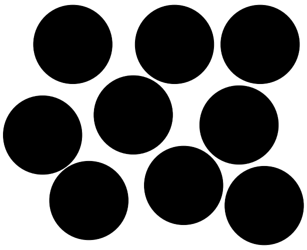
Exponential number of states: spins on a lattice

If you place N spins, constrained to have just two values $S_i = \pm 1$, onto lattice, then the total number of states is proportional to 2^N. Even this simple 5×5 lattice has $2^{25} = 33,554,432$ distinct states. Doubling the length of a side to make a 10×10 lattice, results in over 10^{30} distinct states; the number grows exponentially with lattice size.



Highly peaked distributions: hard spheres near freezing

Hard spheres (HS), round particles that cannot overlap (just like snooker balls), are a popular model for fluids. But even at relatively moderate densities, generating a configuration by assigning random positions to N HS would almost always lead to an overlap which has infinite energy so that the state doesn't contribute to Eq. (1.14). Take, for instance, 100 hard spheres near the freezing transition: only one in about 10^{260} random configurations would count towards the average in Eq. (1.14)[FS2002, p 24]; the distribution is very strongly peaked around a small subset of all possible states.



The simple spin model demonstrates just how rapidly the number of states can grow with the size of the system. Sampling all states is clearly not an option. However, the HS example hints at a way out: Only a small subset of all possible configurations contribute significantly to the average of Eq. (1.14). In fact, these statistical averages are a high dimensional analogue of Fig. 1.3, but with a few much more prominent peaks dominating the integral. If we could somehow sample mainly over states in this set of "peaks", we might still be able to perform an accurate average with a reasonable number of MC sampling points. The way forward clearly involves some form of importance sampling as described in section 1.3. A natural choice would be to sample points from the Boltzmann distribution (1.11), since this is what determines the weight of each state. Applying this to the numerator *and* the denominator of Eq. (1.14), and correcting for the bias as done in Eq. (1.7), cancels the factors p_i to obtain

$$A_M = \frac{\sum_i A_{i/p_i}}{M}. \tag{1.15}$$

The i/p_i reminds us that we sample states over a distribution p_i. This particular importance sampling technique was first described in a seminal paper by Nicolas Metropolis together with the Rosenbluths and the Tellers [M1953]. They calculated the equation of state of hard-discs, and summarised their method in the following words:

> "*So the method we employ is actually a modified Monte Carlo scheme, where, instead of choosing configurations randomly, then weighing them with* $\exp(-E/kT)$, *we choose configurations with a probability* $\exp(-E/kT)$ *and weight them evenly.*"

At first glance this method doesn't appear to be very practical. First of all, although for many systems $\exp[-\beta E_i]$ is relatively straightforward to calculate, the partition function $Q = \sum_i \exp[-\beta E_i]$ is almost always impossible to obtain[6]. By analogy again with Fig. 1.3, it's as if we know how to calculate the relative, but not the absolute height of the peaks. Moreover, the space of states i has such a complex high-dimensional structure, that it is very hard to know a-priori exactly where to look for the most probable states – those with the largest values of $p_i = \exp[-\beta E_i]/Q$ – that dominate the averages. However, in their famous 1953 paper, Metropolis *et al*[M1953] devised a very clever way around these problems. Their method, based on a biased random walk through configuration space, is still by far the most popular technique used in atomic and molecular simulations. The next section explains in more detail how this Metropolis Monte Carlo scheme works.

1.4.2 Efficiently sampling phase space with a random walk

Monte Carlo "trajectories"

Sampling configuration space with a "biased random walk" can be described as follows:

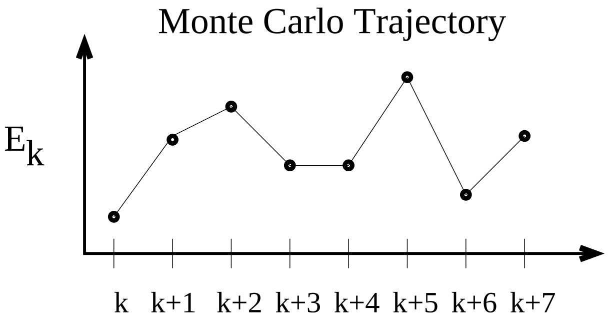
- Start with a given configuration o for which the Boltzmann factor is $\exp[-\beta E_o]$.
- Choose *and* accept a new configuration n, at energy E_n, with a transition probability $\pi(o \to n)$.
- Calculate the value of the operator you are interested in, and add it to your average (1.14)
- repeat to create a MC trajectory through phase space

Note that transition probabilities summed over all other states must add up to unity. This helps define the probability that you stay at the same state o in a MC step:

$$\pi(o \to o) = 1 - \sum_{n \neq o} \pi(o \to n). \tag{1.16}$$

[6] If we could calculate it, we wouldn't need to be doing Monte Carlo!

A Monte Carlo Trajectory:
The Metropolis Monte Carlo algorithm steps through phase space with a random trajectory. If at step k the system is in state o, with energy E_o, then the probability that at step $k+1$ the state will change to n, with energy E_n, is given by $\pi(o \to n)$. The probability that the state at step $k+1$ remains o is given by $\pi(o \to o)$ (see Eq. (1.16)). An example where the state doesn't change would be step $k+3$ to $k+4$. A MC average like Eq. (1.15) should be taken at each step, regardless of whether the state has changed or not.



These MC trajectories are different from the MD trajectories that were discussed by Dr. Sprik. The latter are deterministic, while the former are stochastic, and need not resemble the realistic dynamics of a physical system at all. And, in fact, it is exactly this property of "non-realism" that makes the MC technique so useful: one can invent clever methods that sample phase space much more efficiently – nimbly skipping around bottlenecks – than a realistic dynamics would.

Why detailed balance is important

To implement the Metropolis scheme we need to sample configurations from the Boltzmann distribution: the probability of sampling state o should be given by $P(o) = \exp[-\beta E_o]/Q$. How can this be achieved by MC trajectories?

A useful way to think about this is in terms of an ensemble of many MC trajectories, i.e. a huge number[7] of identical physical systems, but with different random walks through the space of all possible states. Then at any given time we can measure $P(o)$ by counting what fraction of walkers are in state o. Once the system has reached equilibrium then $P(o)$ should be *stationary*: the average population of walkers in any state o shouldn't change with time (i.e. MC steps). This implies that the number of systems making a transition to a given state is equal to the number of systems leaving that state. Expressed in mathematical form this statement becomes:

$$P(o) \sum_i \pi(o \to i) = \sum_j P(j) \pi(j \to o). \tag{1.17}$$

(Before you read on, can you see why both $P(j)$ and $\pi(j \to o)$ are included: what is the difference between the two?) In practise though, a more stringent condition is usually imposed:

$$P(o)\pi(o \to n) = P(n)\pi(n \to o). \tag{1.18}$$

which removes the need for sums and satisfies Eq. (1.17). This is often called **detailed balance**: In equilibrium, the average number of accepted moves from a state o to any other

[7] It's helpful to think of this number as being much larger than the number of states in a given system

state n is exactly cancelled by the number of reverse moves from n to o. Detailed balance guarantees that, once equilibrium is established, the ensemble of random walkers populates the states o with the correct distribution $P(o)$.

In the Metropolis MC method you need to impose a Boltzmann distribution $P(i) \propto \exp(-E/k_BT)$. Eq. (1.18) suggests that to achieve this, all you need to do is choose the correct transition probabilities $\pi(o \rightarrow n)$[8]. It is useful to first split up the determination of $\pi(o \rightarrow n)$ into two steps:

1. Choose a new configuration n with a *transition matrix* probability $\alpha(o \rightarrow n)$.

2. Accept or reject this new configuration with an *acceptance probability* $acc(o \rightarrow n)$.

In other words, the transition probability has been rewritten as:

$$\pi(o \rightarrow n) = \alpha(o \rightarrow n) acc(o \rightarrow n) \tag{1.19}$$

Many MC methods take α to be symmetric, i.e. $\alpha(o \rightarrow n) = \alpha(n \rightarrow o)$. The detailed balance condition (1.18) therefore implies that:

$$\frac{\pi(o \rightarrow n)}{\pi(n \rightarrow o)} = \frac{acc(o \rightarrow n)}{acc(n \rightarrow o)} = \frac{P(n)}{P(o)} = \exp\left[-\beta\left(E_n - E_o\right)\right], \tag{1.20}$$

where only the last equal sign used the fact that the $P(i)$ should follow the Boltzmann distribution. By choosing transition probabilities $\pi(o \rightarrow n)$ in this way, which conserves detailed balance, the equilibrium population of MC trajectories will populate the states with the desired Boltzmann distribution. One very clever aspect of this scheme is that there is no need to ever directly evaluate the partition function Q!

1.4.3 The Metropolis Algorithm

Acceptance criteria for the Metropolis Algorithm

There are many possible choices of $acc(o \rightarrow n)$ that would satisfy detailed balance and condition 1.20. Here we only discuss the algorithm of Metropolis *et al.*, which, 50 years after its introduction, is still by far the most popular recipe. Their inspired choice was:

$$\begin{aligned} acc(o \rightarrow n) &= P(n)/P(o) = \exp[-\beta(E_n - E_o)] \quad \text{if} \quad P(n) < P(o) \\ &= 1 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if} \quad P(n) \geq P(o) \end{aligned} \tag{1.21}$$

which means you always accept the trial move if energy decreases, and you accept energy increases with the Boltzmann factor of the energy difference. At first sight the two different scenarios for $acc(o \rightarrow n)$ may seem strange because they are asymmetric. However, a probability can't be larger than 1, and you should be able to convince yourself that plugging this condition into Eq. (1.20) satisfies detailed balance, and thus leads to the correct Boltzmann distribution of random walkers. (Can you think of other choices for $acc(o \rightarrow n)$ that satisfy Eq. (1.20)?)

[8] In principle we could also use Eq. (1.17) to choose the $\pi(o \rightarrow n)$ that generate a Boltzmann distribution, but the sums over all states make this more general condition much harder to use than the simpler detailed balance condition (1.18).

Importance sampling: To work out what percentage of yesterday's travellers on the tube slept during their journey, you could sample randomly (or uniformly) over the entire area where you might find them – lets say you're only interested in those who live in England and Wales – or you could do a biased sampling with a much higher probability to test in London, and so gather your data more efficiently. This is the essence of "importance sampling".

Sampling by a drunkards walk in London: But what happens if you don't know the extent of (greater) London? This is analogous to the situation encountered when calculating the (very!) high dimensional integrals needed for statistical mechanics. The method of Metropolis *et al.* solves this problem by using a random (or drunkards) walk algorithm. It works roughly like this: First, start with someone who took the tube yesterday. Ask them **question 1**: *did you sleep on the tube yesterday?* to begin your averaging. Then repeat the following process: Take a big step (or steps) in a random direction. Ask the first person you then see **question 2**: *did you take the tube yesterday?*. If they say yes, ask them question 1, and add the result to your average. Then take new steps in a random direction and repeat. If, on the other hand, they say no to question 2, go back to your original location, and add the original answer to question 1 to your average again. Then take a new random step in a different direction etc...

Here the size and direction of your steps corresponds to the transition matrix $\alpha(o \rightarrow n)$, and **question 2** corresponds to the acceptance probability $acc(o \rightarrow n)$. In this way you are generating your importance sampling distribution, which is zero for people who didn't take the tube, but finite for those who did. **Question 1**: *did you sleep on the tube yesterday?* is used for averaging, much like is done in Eq. (1.14).

Clearly you need to start somewhere near London to have any chance for your method to work (if you start in the Welsh countryside, you may be stuck there forever). You may also need to adjust the length or number of your steps to optimise your sampling.

The population of people, living in England and Wales, who took the tube yesterday is mainly peaked around London. If you take random samples over all of England and Wales, somewhat like the dots in this picture, you are quite unlikely to get a good measure of what percentage of those who took the tube fell asleep during their journey

Applying the Metropolis algorithm to a simple fluid

Now that we've finally derived the Metropolis algorithm, let's illustrate it with a more concrete example. Consider a fluid of N particles with positions described by $\mathbf{r}^N = \{\mathbf{r_i}\} = (\mathbf{r_1}, \mathbf{r_2}, ...\mathbf{r_N})$, interacting through a potential $\mathcal{V}(\mathbf{r}^N)$. In the discussion up to now we've always considered the energy E_i of state i, which normally includes both kinetic and the potential energy contributions. But, as shown in section 1.2.2 of Dr. Sprik's lecture notes on the MD method, the kinetic energy is a rather trivial quantity in classical statistical mechanics[9]. We can easily integrate over the momenta in the partition function to find, as shown in Eq. 1.45 of those notes, that $Q_N = (N!\Lambda^{3N})^{-1} Z_N$, where Λ is the thermal wavelength, and

$$Z_N = \int_V d\mathbf{r}^N \exp\left[-\beta \mathcal{V}\left(\mathbf{r}^N\right)\right] \tag{1.22}$$

is the configurational integral. Since the integral over momentum just gives a constant factor that cancels between the numerator and denominator of Eq. (1.14), we can ignore the momentum and do our MC averages over states that are determined by $\mathbf{r}^N$ alone. In other words, each new set $\mathbf{r}^N$ of fluid particle positions corresponds to a new state i of the system, and the probability distribution $p_i = \exp[-\beta E_i]/Q$ reduces to the following form:

$$P(\mathbf{r}^N) = \frac{1}{Z_N} \exp\left[-\beta \mathcal{V}\left(\mathbf{r}^N\right)\right]. \tag{1.23}$$

In the next box, we summarise the algorithm that Metropolis *et al.*[M1953] introduced for the MC evaluation of thermodynamic averages for a fluid.

[9] as opposed to quantum mechanics where it can be very hard to calculate

Summary of Metropolis Algorithm for fluids

To move from step k to step $k+1$ in the Monte Carlo trajectory repeat the following:

1 select a particle j at random from the configuration of step k:

$$\mathbf{r}_k^N = (\mathbf{r_1}, ..., \mathbf{r_j}, ..\mathbf{r_N}).$$

2 move it to a new position with a random displacement $\mathbf{r}'_j = \mathbf{r}_j + \mathbf{\Delta}$

3 Calculate the potential energy $\mathcal{V}(\mathbf{r}^N_{trial})$ for the new trial state $\mathbf{r}^N_{trial} = (\mathbf{r_1}, ..., \mathbf{r}'_\mathbf{j}, ...\mathbf{r_N})$

4 Accept the move $\mathbf{r}_k^N \to \mathbf{r}^N_{trial}$ with a probability

$$acc(o \to n) = min\left\{1, \exp\left[-\beta\left(\mathcal{V}(\mathbf{r}^N_{trial}) - \mathcal{V}(\mathbf{r}^N_k)\right)\right]\right\}$$

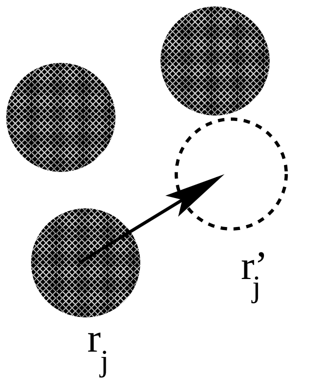
If the trial move is accepted then the state at step $k+1$ is given by

$$\mathbf{r}^N_{k+1} = \mathbf{r}^N_{trial};$$

if it is not accepted then

$$\mathbf{r}^N_{k+1} = \mathbf{r}^N_k.$$

Note that trial moves to lower potential energy are always accepted, whereas moves to a higher potential energy are accepted with a finite probability $\exp\left[-\beta\left(\mathcal{V}(\mathbf{r}^N_{trial}) - \mathcal{V}(\mathbf{r}^N_k)\right)\right]$ that decreases for increasing energy difference.



Trial move of a particle at $\mathbf{r}_j$ to $\mathbf{r}'_j$

5 Add the value of your operator at step $k+1$, i.e. $A(\mathbf{r}^N_{k+1})$, to the average

$$A_{k+1} = \frac{1}{k+1}\sum_{i=1}^{k+1} A(\mathbf{r}^N_i)$$

Go back to step 1 and repeat.

A few comments on the schematic outline above:

- Steps 1 and 2 together define the transition matrix $\alpha(o \to n)$ of Eq. (1.19) with $o = \mathbf{r}^N_k$ and $n = \mathbf{r}^N_{trial}$. Note that it is symmetric, i.e. $\alpha(o \to n) = \alpha(n \to o)$, as required.
- If the random displacement Δ is too large, then most trial steps will be rejected, while if Δ is too small you will only move very slowly through phase-space. The optimum Δ, which leads to the most efficient sampling of phase-space, should be somewhere in between. For many systems a good rule of thumb is to choose an average acceptance of about 50%[10]. For a dense liquid the optimum Δ will be smaller than for a dilute

[10] This works best for systems with continuous potentials, where the calculation of $\mathcal{V}(\mathbf{r}^N)$ is an expensive step. For systems like hard spheres, a larger Δ, leading to a lower acceptance probability, is often more efficient because you can reject a move as soon as you find the first overlap. A more general way to estimate

fluid. (Can you think why this is so?). Since it's hard to tell a-priori what the optimal Δ will be, it is often adjusted once the MC simulation is under way.

- At each MC step only one particle is moved, so that N separate MC steps are roughly equal in cost to a single Molecular Dynamics step, where all particles are moved together according to Newton's equations of motion. You might wonder why we don't also move all N particles together in a single MC step. The reason is quite simple: the main cost in a MC algorithm is usually evaluating $\mathcal{V}(\mathbf{r}^N)$ for a new configuration. Suppose that we can truncate the interactions so that the cost of moving any single particle scales as its average number of neighbours m. The cost of N single moves then scales as mN, which is similar to the cost of a single N particle move. However, if the probability of getting an overlap (and a rejected step) when moving a single particle by a distance Δ is p_{rej}, then the probability of *accepting* a collective move of all N particles scales roughly as $(1-p_{rej})^N$. To get any acceptances at all p_{rej} would have to scale as $1/N$, which implies an extremely small step Δ. In other words, for roughly the same amount of computational work (i.e. CPU time), a single particle move algorithm advances each particle much further on average than a collective move algorithm does. This is why most MC programs mainly use single-particle moves[11].

1.4.4 Pseudo Code for the MC Algorithm

In this section we use a simplified pseudo-code to describe a MC algorithm for a fluid interacting through a pair potential. (Here I borrow liberally from FS2002)

```
Program MC                              Basic Metropolis MC Algorithm

  do icycl:= 1, Ncycle                  perform Ncycle MC cycles
        call MCmove                     accept or reject a trial move
        if(mod(icycl,Nsample)=0) then
              call sample               sample averages every Nsample steps
        endif
  enddo
end
```

description of the subroutines

The subroutine *MCmove*, described in the box below, attempts to displace a random particle The subroutine *sample* adds to averages of the form Eq. (1.14) every *Nsample* steps

the optimum Δ for the most efficient sampling of phase-space is to maximise the ratio of the sum of the squares of the accepted displacements divided by the amount of computer time.

[11] Later in the course we describe situations where adding some clever collective moves does lead to a more efficient sampling of phase space.

```
subroutine MCmove                                  A routine that attempts to move a particle
    o = int(ran(iseed) ∗ npart) + 1                select one of npart particles at random
   call energy(xo,yo,zo,Eo)                        energy Eo of the old configuration
    xn = x0 + (ran(iseed)-0.5) ∗ delx              give a particle a random x displacement
    yn = y0 + (ran(iseed)-0.5) ∗ dely              give a particle a random y displacement
    zn = z0 + (ran(iseed)-0.5) ∗ delz              give a particle a random z displacement
    call energy(xn,yn,zn,En)                       energy En of the new configuration
    if (ran(iseed) < exp(-beta*((En-Eo))) then     acceptance rule 1.21
            xo = xn                                replace xo by xn
            yo = yn                                replace yo by yn
            xo = zn                                replace zo by zn
    return
end
```

1.5 Asides on ergodicity and Markov chains

Quite a few assumptions and subtleties were swept under the carpet in the derivation of the Metropolis algorithm described above. Here we'll mention a few of them, some which could have implications for a practical MC algorithm.

- The biased random walk is more formally known as a *Markov process.* This is a general name for stochastic processes without "memory", i.e. the probability to make a step from state o to state n is independent of the previous history of steps.

- A very important assumption is that of *ergodicity.* In brief, this implies that our Markov process can move from any one state i to any other state j in a finite number of steps. Systems with "bottlenecks" in phase-space will often cause problems. Special techniques which mix different kinds of moves can sometimes help overcome this problem.

- The Markov chain should not be periodic. If, for example, you have a two component system, and your algorithm to choose a particle always alternates between species 1 and species 2, then at any given step you always know which species you started from, and you may not sample correctly.

- Under fairly general conditions, it can be proven that a Markov chain approaches the equilibrium distribution exponentially fast. Nevertheless in practise, this can still take quite a few MC steps, and this may also depend very strongly on your starting configuration. When you first start a MC program, you have to *equilibrate* your system – run it until you are satisfied that the correct distribution has been reached – before collecting averages. Equilibration errors are very common[12].

[12] In fact, even Metropolis *et al.* [M1953], probably didn't equilibrate long enough

Example: Equilibration for the Ising Model

The 2-D Ising model, which you saw in Part II practicals, has a Hamiltonian of the following form:

$$H = -\frac{J}{k_B T}\sum_{<i\ j>} S_i S_j + H\sum_i S_i \tag{1.24}$$

where the spins $S_i = \pm 1$, and the double sum is over all distinct pairs. It can be viewed as a very crude model of a magnet. When the external field $H = 0$, then for positive J, the spins can lower their energy by aligning with their nearest neighbours. However, this process competes with entropy, which is maximised when the spins are disordered. For low enough temperature ($k_b T/J \leq 2.29$), this entropy loss is overcome, and the system can spontaneously break its symmetry so that on average the spins will point up or down, behaviour resembling ferromagnetism. A small external field H will then set the direction of the spins, which can be measured by the *magnetisation* M which is defined for an $N \times N$ lattice by

$$M = \frac{1}{N^2}\sum_i S_i. \tag{1.25}$$

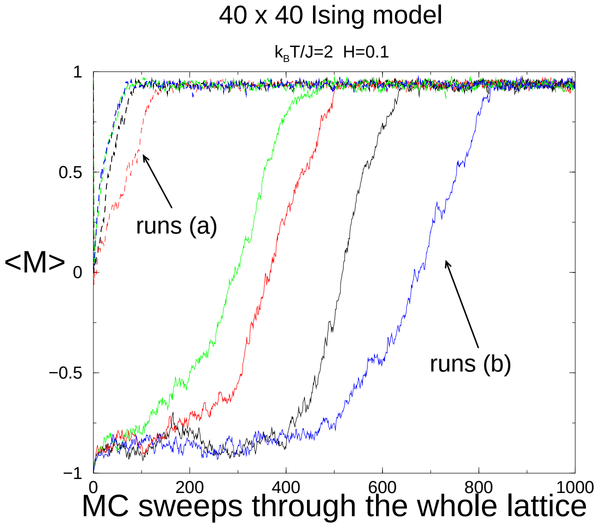
When $H > 0$ then $M > 0$ and we say the spins are pointing up on average, while if $H < 0$ then $M < 0$ and the spins point down on average. We saw earlier how fast the number of states grows with the number of spins; for this reason MC techniques are often employed to study the behaviour of the Ising model[13].

The figure on the right shows the equilibration behaviour of a simple Metropolis MC simulation of a 40×40 Ising model with periodic boundary conditions. Here $k_B T/J = 2$, which is below the transition temperature, and the external field is set to $H = 0.1$, which favours a configuration with most spins pointing up. Two initial conditions were used:

(a) random spin configuration (dashed lines)

(b) all spins down (solid lines)

with 4 independent runs each.



Here are some lessons you can immediately infer from the graph:

- Even for the same initial conditions, equilibration times can vary from run to run.
- Good initial conditions accelerate the approach to equilibrium.

[13] In 2-dimensions there exists an exact solution by the famous physical chemist Lars Onsager (Phys. Rev. **65**, 117 (1944)), but it is valid only for for $H = 0$. In spite of the simplicity of the model, it has so far resisted all attempts at a solution in 3-D.

– Averages should only be taken after the system has reached equilibrium.

1.6 Estimating statistical errors

Steps in MC trajectories are not statistically independent

The variance of an M step MC simulation (measured after equilibrium has been reached), is given by

$$\sigma_M^2 = \frac{1}{M} \sum_{k=1}^{M} (A_k - < A >_M)^2 =< A^2 >_M - < A >_M^2 \tag{1.26}$$

If the M measurements of A_k were truly independent we could estimate the variance in the average $< A >_M$ by:

$$\sigma^2(< A >_M) \approx \frac{1}{M} \sigma_M^2, \tag{1.27}$$

However, this would be incorrect because the MC trajectories are *correlated*, i.e. it takes a number of steps to move from one independent part of phase-space to another. This rather vague statement can be made a bit more precise by calculating the auto-correlation function:

$$C_{AA}(k) = \frac{1}{M} \sum_{k'} (A_{k'} - < A >) (A_{k'+k} - < A >) \tag{1.28}$$

which measures how long it takes for the system keeps a “memory” of what state it was in. This correlation function typically exhibits an exponential decay:

$$C_{AA}(k) \sim \exp(-k/n_\tau) \tag{1.29}$$

and $2n_\tau$ is often taken as the number of steps there are between independent measurements. Therefore, for a MC trajectory of M steps, only

$$n_M = \frac{M}{2n_\tau} \tag{1.30}$$

can be considered to be statistically independent measurements. A better estimate for the variance in the average would be:

$$\sigma^2(< A >_M) = \frac{1}{n_M - 1} \sigma_M^2 = \frac{1}{n_M - 1} \left(< A^2 >_M - < A >_M^2\right) \tag{1.31}$$

which assumes n_M independent measurements[14]. Note that this estimate may be much larger than what you get from naively assuming that all MC steps are independent measurements.

[14] Note that we used the more accurate $1/(n_M - 1)$ factor instead of the more commonly used $1/n_M$ (see any good book on error estimates for how to derive this.). For large n_M the differences are negligible, but this may not always be the case for many realistic MC simulations, where n_M can be rather small.

Estimating errors with block averages

Another way to estimate the correct variance, which circumvents the need to calculate a correlation function and the concomitant subtleties in extracting n_τ, is to use *block averages.* In brief the method works like this: Once you have your sequence of M measurements of the fluctuating variable A, take L partial averages $< A >_l$ over blocks of length $l = \frac{M}{L}$ Monte Carlo steps each. The variance $\sigma_L^2(A)$ can then be measured as:

$$\sigma_L^2(< A >) = \frac{1}{L}\sum_{i=1}^{L}[A_l - < A >]^2 \tag{1.32}$$

where $< A >$ is the original average over all M steps. As the block size is increased, eventually $L \leq n_M$, so that the measurements become independent, and the laws of statistics predict:

$$\frac{\sigma_L^2(< A >)}{L - 1} \approx \sigma^2(< A >). \tag{1.33}$$

$\sigma^2(A)$ is the true variance of average of the fluctuating variable A, measured for N particles in whatever ensemble you used. To achieve this in practise, take your data and plot Eq. (1.33) as a function of L. It should grow with decreasing L, and plateau at roughly the correct variance for $L \leq n_M$[15].

There are quite a few subtleties in calculating errors in MC. For example, even in the same simulation, different properties may have different correlation "times" n_τ. Moreover, there are important differences with how single particle and collective variables behave. See e.g. appendix D of FS2002 for a good discussion of some of these issues.

Systematic errors

Besides statistical errors, which are more straightforward to quantify, a MC simulation can have *systematic errors.* Two of the most common are:

- **finite size errors**

  A MC simulation will, by necessity, always be on a finite size system. If you are interested in thermodynamic averages, these are usually defined for infinitely large systems. The so-called finite size error for many simple properties scales roughly as:

  $$\text{Error} \sim \frac{1}{\sqrt{N}}, \tag{1.34}$$

  where N is the number of particles. Even if you were to do an infinite number of MC steps, you would not converge to the same result as for an infinite size box. In practise, if you are interested in the thermodynamic limit, you can run your simulation for several different numbers of particles, and extrapolate to $N = \infty$, a procedure called *finite size scaling.* However, you do need a good idea of how the property you are interested in scales with system size, which may not always be as simple as Eq. (1.34).

[15] Once L becomes too small, then the fluctuations in the variance (errors in the error) become relatively large, making it hard to see the plateau. If your simulation was long enough, however, there should still be an intermediate set of L that produces a reliable plateau.

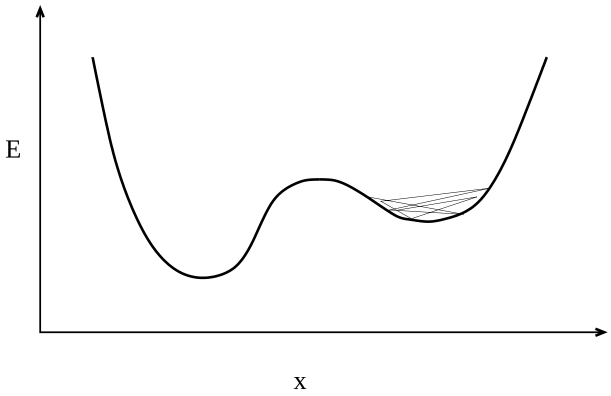


Figure 1.4: Schematic of a simulation stuck in a metastable minimum. This MC simulation trajectory remains in the same energy well, even though it is not the lowest energy state.

Besides these statistical errors, finite size simulations may also introduce more serious systematic errors. For example, properties which depend on fluctuations with wavelengths larger than the smallest length of your simulation box will not be sampled properly. The properties of phase-transitions usually show important finite size effects. In general, it is a good idea to test your simulation on several different system sizes until the property you are studying no longer varies.

- **equilibration errors**

  The simplest equilibration errors occur when averages are taken before the system has reached equilibrium. One obvious way to check this is to monitor the variables you are interested in, and only take averages after they have stopped "drifting". But a more serious problem occurs when the system is stuck in a local minimum of the energy landscape, as Fig. 1.6 shows. Then the system may appear to equilibrate quite nicely inside the local well, even though it is not sampling phase-space correctly (this could also be classed as an ergodicity problem). The runs (b) in the Ising model example show this behaviour: The initial conditions have all spins down while the lowest energy state has all spins up. If you sampled for a short amount of time, (say less than 100 MC sweeps) before the spins collectively flip over, you might erroneously think that the system had equilibrated[16]. As illustrated in that example, where the random configuration equilibrates much faster, it's often a good idea to perform a simulation with several different starting configurations.

[16] In fact, for this particular system, the time to equilibration grows exponentially with decreasing k_BT/J

Chapter 2

Calculating thermodynamic properties with Monte Carlo

2.1 Brief reminder of the Statistical Mechanics of ensembles

The canonical ensemble (fixed N, V, T) is the natural choice for simple MC calculations, just as the microcanonical ensemble (fixed N, V, E) is the natural one for MD simulations. Experiments, on the other hand, are typically performed at constant pressure P. Moreover, considerations of computational efficiency may impose the constraint of other parameters. For example, fixing the chemical potential μ is often useful when studying adsorption processes.

The inherent flexibility of MC moves makes this technique uniquely suited for sampling other ensembles. But before describing details of implementation, we will engage in a "climb up", à la Feynman, to a few related summits, and briefly discuss some of the more popular ensembles, summarised in the table below [1].

Summary of 4 popular ensembles

Ensemble	Thermodynamic potential	Partition function	Probability distribution
microcanonical fixed (NVE)	$S/k_B = \log[\Omega(N, V, E)]$ entropy	$\Omega = \sum_i \delta_{E,E_i}$	$p_i = \frac{1}{\Omega}$
canonical fixed (NVT)	$-\beta A = \log[Q(N, V, T)]$ Helmholtz free-energy	$Q = \sum_i \exp[-\beta E_i]$	$p_i = \frac{\exp[-\beta E_i]}{Q}$
isobaric fixed (NPT)	$-\beta G = \log[\Delta(N, P, T)]$ Gibbs free energy	$\Delta = \sum_i \exp[-\beta(E_i + PV_i)]$	$p_i = \frac{\exp[-\beta(E_i + PV_i)]}{\Delta}$
grand canonical fixed (μVT)	$\beta PV = \log[\Xi(\mu, V, T)]$ Grand potential	$\Xi = \sum_i \exp[-\beta(E_i - \mu N_i)]$	$p_i = \frac{\exp[-\beta(E_i - \mu N_i)]}{\Xi}$

[1] Later sections will hopefully convince you that computer simulations are more than just a slide down towards applications – they often bring into focus the subtle differences between ensembles.

microcanonical

The microcanonical ensemble is in a sense the most basic one. Three extensive variables, the total energy E, particle number N, and volume V are fixed. All states are equally likely, i.e. $p_i = \frac{1}{\Omega}$, so that the partition function $\Omega(E, V, T)$ is a simple sum of the total number of microstates of the system at those fixed parameters. The link between the partition function and thermodynamics proceeds through the entropy[2] $S/k_B = \log[\Omega(E, V, T)]$ from which the thermodynamic definition of the temperature

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E}\right)_{V,N} \tag{2.1}$$

and other thermodynamic properties can be derived.

However, this ensemble is not well suited to Monte-Carlo calculations, because each state is equally probable and the advantages of importance sampling over a small subset of states is lost.

canonical

The canonical ensemble can be derived by imagining a microcanonical system, split into a smaller (but still macroscopic) subsystem I with energy E_i, V_i, N_i, and a larger subsystem II with $E - E_i, V - V_i, N - N_i$, as depicted in Fig. 2.1. The borders between I and II keep V_i and N_i fixed, but allow energy to be shared between the two subsystems. By taking the limit where the size of system II relative to system I goes to infinity – ($(N - N_i)/N_i \to \infty$ with $\rho = N_i/V_i = (N - N_i)/(V - V_i)$ fixed – but still keeping the smaller system macroscopic), system II effectively becomes microcanonical again. The probability of finding the smaller subsystem with an energy E_i is equal to finding the larger system with energy $E - E_i$. The latter can now be viewed in the microcanonical ensemble, so the probability is given by:

$$p_i = \frac{\Omega(E - E_i)}{\sum_j \Omega(E - E_j)} \tag{2.2}$$

where the $\Omega(E - E_i)$ are the microcanoncial partition functions of subsystem II. Since we are treating the limit where system I is much smaller than system II, it makes sense to expand $\log[\Omega(E - E_i)]$ (which, in contrast to $\Omega(E)$, is extensive), in the small parameter $x = E_i/E$ around $x = 0$ which gives:

$$\begin{aligned} \log[\Omega(E - E_i)] &= \log[\Omega(E)] + x\left(\frac{\partial \log[\Omega(E(1-x))]}{\partial x}\right)_{x=0} + \mathcal{O}(x^2) \\ &\approx \log[\Omega(E)] - \frac{E_i}{k_B T} \end{aligned} \tag{2.3}$$

where in the last line we've changed variables from x to E_i, and used the relationship $S/k_B = \log[\Omega(E, V, N)]$, and the definition of temperature, Eq. (2.1), to simplify. Using Eq. (2.3) in Eq. (2.2) then gives

$$p_i = \frac{\Omega(E)\exp[-\beta E_i]}{\sum_j \Omega(E)\exp[-\beta E_j]} = \frac{\exp[-\beta E_i]}{\sum_j \exp[-\beta E_j]}. \tag{2.4}$$

[2] This relationship between entropy and the number of states in a system, which Ludwig Boltzmann had inscribed on his tombstone, could also make a strong claim to being the summit of statistical mechanics.

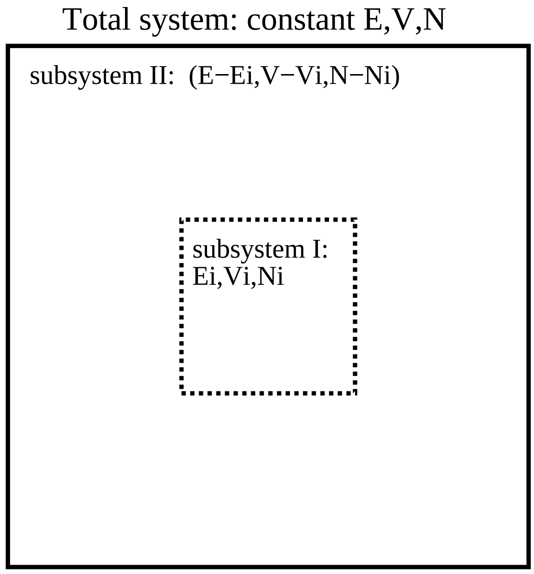


Figure 2.1: Defining a subspace of a larger microcanonical system helps in deriving different ensembles.

This little "climb up to the summit" helps explain why in the canonical ensemble the p_i decrease with increasing E_i: system I is coupled to a much larger system for which the number of states peaks at $x = 0$[3].

Since the energy is allowed to fluctuate, the subsystem takes on the temperature T of the total system, which can be viewed as a heat bath. Thus the relevant parameters for the canonical partition function $Q(N, V, T)$, defined by the denominator in Eq. (2.4), are two extensive variables, N and V, and the intensive variable T. The connection to thermodynamics proceeds via the Helmholtz free energy $\beta A = -\log\left[Q(N, V, T)\right]$

isobaric

The isobaric ensemble can be derived in a similar fashion to the canonical ensemble. Now the subsystem I of Fig. 2.1 allows not only energy, but also volume to fluctuate[4]. This means that there is one fixed extensive variable, N, and two intensive variables, the temperature T and the pressure P. The partition function $\Delta(N, P, T)$ and phase-space probabilities are given in the table, while the connection to thermodynamics proceeds via the Gibbs free energy $\beta G = -\log\left[\Delta(N, P, T)\right]$. Note that the sum over states includes not only different energies, but also different volumes.

[3] see e.g. FS 2002, p 9-13, or a number of standard texts on statistical mechanics for a more detailed derivation.

[4] It's easier to derive this ensemble by first following the subsystem procedure to define a canonical ensemble, and then repeating it with walls that allow volume fluctuations (like a big piston) to derive the isobaric ensemble

grand canonical

To derive this ensemble simply define an imaginary boundary for system I. The thermodynamic states are determined by the (fluctuating) subset of particles within the prescribed volume. Particles, and with them energy, freely move across the imaginary borders of the box, leading to two constant intensive variables, the temperature T and the chemical potential μ. The required extensive variable is the volume V[5]. This exactly describes the grand-canonical ensemble. Expressions for the partition function $\Xi(\mu, V, T)$, the probability p_i, and the connection to thermodynamics through $\beta PV = \log\left[\Xi(\mu, V, T)\right]$ are given in the table.

averages and fluctuations

The number of particles in a canonical ensemble is fixed at N, whereas in a grand-canonical ensemble N differs from state to state, with relative fluctuations of order $1/\sqrt{N}$. The converse holds for the chemical potential μ. However, for the same system at the same state point, the ensemble averages of each quantity will be equal, at least in the thermodynamic limit. For example, if for a given N, V, T you measure $\mu = \langle\mu\rangle_{N,V,T}$ in the canonical ensemble, then in the grand canonical ensemble, for the same μ, you will find $\langle N\rangle_{\mu VT} = N$[6].

It is important to remember that this equivalence of averages does not hold for *fluctuations*. For some variables this is obvious; consider for example fluctuations in the energy E. Whereas these will be zero (by definition) in the microcanonical ensemble, they will be finite in the other three ensembles discussed above. Fluctuations can often be related to thermodynamic properties, and are therefore useful quantities in computer simulations. But care must always be taken in their interpretation. For example, in the canonical ensemble, the fluctuations in the total energy can be directly related to the specific heat $C_V = (\partial E/\partial V)_{NT}$ at constant volume:

$$\left\langle (E - < E >)^2 \right\rangle_{NVT} = k_B T^2 C_V. \tag{2.5}$$

But this does not hold for energy fluctuations at constant NPT (the isobaric ensemble). Nevertheless, for such systems, the fluctuations in the instantaneous enthalpy $H = E + PV$ can be used to calculate the specific heat at constant pressure:

$$\left\langle (H - < H >)^2 \right\rangle_{N,P,T} = k_B T^2 C_P \tag{2.6}$$

A whole host of other useful fluctuation relations exist, and can be found in standard texts[7]. The take home message is that one should always be careful to use the appropriate fluctuation relationship for the ensemble one is calculating with.

[5] Can you think of reasons why defining an ensemble with three intensive variables causes difficulties for simulations?

[6] Note that this is strictly true only in the thermodynamic limit. In practise a simulation is always for a finite sized system, and the finite size effects may not be exactly the same in each ensemble.

[7] Such as FS2002, or Allen and Tildesley 1987

2.2 Constant pressure MC

Experiments are often naturally performed at constant pressure, so simulations in this ensemble can be quite useful. How does one go about implementing the isobaric ensemble in a MC computer simulation? To answer that, we first specialise to a classical fluid, where, just as for the canonical ensemble, as described in in Eqs. (1.22) and (1.23), the momentum coordinates can be intergrated out of the partition function Δ defined in the table. Averages then take the form:

$$\langle A \rangle_{NPT} = \frac{\int_0^\infty dV \exp\left[-\beta PV\right] V^N \int d\mathbf{s}^N A(\mathbf{s}^N; V) \exp\left[-\beta \mathcal{V}(\mathbf{s}^N; V)\right]}{Z(NPT)} \tag{2.7}$$

where $Z(NPT)$ is generalisation of the configurational integral $Z_N = Z(N, V, T)$ defined in Eq. (1.22) for the canonical ensemble. It is the function that would normalise the distribution. Here we've scaled the particles in the configuration $\mathbf{r}^N = (\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_N)$ by the volume of the box – each $\mathbf{s}_j$ in $\mathbf{s}^N$ is given by $V^{-\frac{1}{3}}\mathbf{r}_j$ – to make explicit the dependence of $d\mathbf{r}^N$ on the volume (This is the origin of the V^N term in Eq. (2.7)). In analogy to Eq. (1.23), valid for the canonical ensemble, a Monte-Carlo algorithm for the NPT ensemble should sample states with a probability distribution proportional to:

$$P_{NPT}(\mathbf{s}^N, V) \propto \exp\left[-\beta\left(\mathcal{V}(\mathbf{s}_N) + PV - N\beta^{-1}\log\left[V\right]\right)\right] \tag{2.8}$$

But now states are not only defined by the configuration $\mathbf{s}^N$, but also by the volume V, which can fluctuate. To properly sample this distribution we need two kinds of MC moves:

1. moves that randomly displace a particle

2. moves that randomly change the volume V

Moves of type 1 were described in chapter 1. Moves of type 2 can also be performed by an adaptation of the Metropolis prescription: The transition probability $\pi(o \to n) = \alpha(o \to n)acc(o \to n)$ is again split into a symmetric transition probability $\alpha(o \to n)$ (the probability to choose a certain trial volume move from V_o to $V_n = V_o + \Delta V$) and an acceptance probability $acc(o \to n)$ chosen such that the MC trajectory reproduces the distribution $P_{NPT}(\mathbf{s}^N, V)$ of Eq. (2.8):

$$acc(o \to n) = min\left\{1, \exp\left[-\beta\left(\mathcal{V}(\mathbf{s}^N; V_n) - \mathcal{V}(\mathbf{s}^N; V_o) + P(\Delta V) - N\beta^{-1}\log\left[\frac{V_o + \Delta V}{V_o}\right]\right)\right]\right\}. \tag{2.9}$$

Even though $\mathbf{s}^N$ itself remains constant, the change of volume scales all the coordinates, and therefore affects the potential energy $\mathcal{V}(\mathbf{r}^N; V)$[8].

[8] It is instructive to examine the case of an ideal gas, where $\mathcal{V}(\mathbf{r}^N) = 0$: the log $\left[\frac{V_o+\Delta V}{V_o}\right]$ and the $P(\Delta V)$ terms work in opposite directions. Without the log term, any move which shrunk the volume would be accepted, leading to a collapse of the system.

volume moves

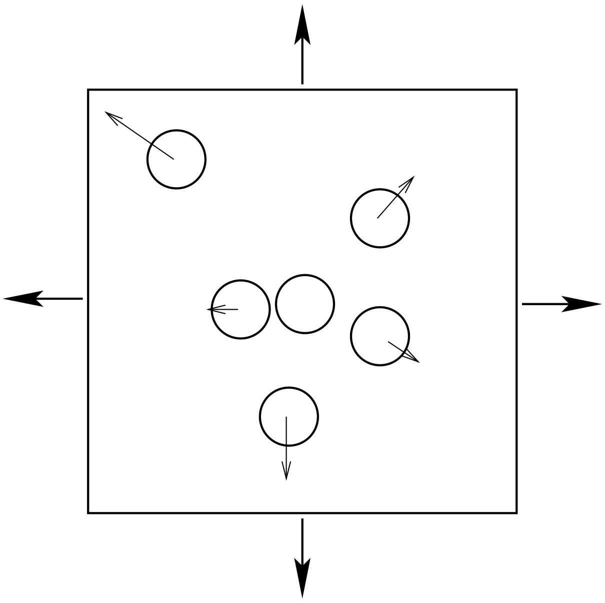
The cost of a volume move depends very much on the type of potentials used. For interactions of the type:

$$\mathcal{V}(\mathbf{r}^N) = \sum_{i<j} \epsilon \left(\frac{\sigma_{ij}}{r_{ij}} \right)^m \qquad (2.10)$$

the effect of a volume change is very simple:

$$\mathcal{V}(\mathbf{s}^N, V_n) = \left(\frac{V_o}{V_n} \right)^{m/D} \mathcal{V}(\mathbf{s}^N, V_o), \qquad (2.11)$$

where D is the system dimension. For these potentials, or those made up of sums of such terms (like the Lennard Jones form), a volume move is inexpensive. For other types of interactions a volume move can mean recalculating $\mathcal{V}(\mathbf{r}^{\mathcal{N}})$ for all the new distances, which is roughly as expensive as N separate one-particle MC moves.



Upon a volume move from V_o to V_n (here $V_n > V_o$), in 3 dimensions, all molecule centre of mass distances from the origin are scaled by a factor $(V_n/V_o)^{\frac{1}{3}}$. Note that particles near the edges move further than particles near the origin.

The cost of a volume move depends on the type of potential; for potentials with one length-scale it will be cheap (see box above), but for more complicated potentials, such as those often used in molecular simulations, the move will be expensive. In the former case, volume moves can be attempted as often as particle moves, while in the latter case, they should be attempted much less often, perhaps once every N particle moves. Either way, it is important make the choice of when to take a volume move randomly, and not after a fixed number of particle moves. Otherwise we could violate detailed balance, which states that for any move $o \rightarrow n$ we should also be able to make the move $n \rightarrow o$. Just as for particle moves, one should optimise the average size of a volume move. Choose it too small and you will find many acceptances, but creep through phase-space very slowly; choose it too large, and the number of accepted moves will be extremely low, which is also computationally inefficient.

```
program MCnpt                                  isobaric ensemble Monte Carlo program

   do mcycl := 1,ncycl                         do ncycl moves
        i := int((natom + 1)* ran(iseed))+1
        if (i ≤ natom) then                    chooses volume move only once in natom tries
        MCmove                                 perform a particle move
        else
        VolMCmove                              perform a volume move
        endif
   enddo
   end
```

2.3 Grand Canonical MC

In the grand-canonical ensemble, particles can be exchanged with a “particle bath”, which fixes the chemical potential μ. For a classical fluid, the probability distribution is proportional to

$$P(\mathbf{s}^N; N) \propto \frac{\exp\left[\beta\mu N\right] V^N}{\Lambda^{3N} N!} \exp\left[-\beta \mathcal{V}(\mathbf{s}^N)\right] \tag{2.12}$$

and a state is defined not only by the particle configurations, but also by the number of particles N. Besides the usual particle displacement moves, we now also need particle insertion or removal moves. Again the transition probability $\pi(o \rightarrow n)$ can be split up into a transition matrix, $\alpha(o \rightarrow n)$ and an acceptance probability $acc(o \rightarrow n)$. Besides the usual particle moves, there are now two additional trial moves: 1) removing a randomly chosen particle and 2) inserting a particle at a random position. It is convenient to simply set the probability of attempting a trial insertion or removal to be equal, i.e. to make the transition matrix symmetric: $\alpha(N \rightarrow N+1) = \alpha(N+1 \rightarrow N)$. In that case we only need to determine the acceptance probabilities, given by:

1. The removal of a particle, chosen at random

$$acc(N \rightarrow N-1) = min\left\{1, \frac{\Lambda^3 N}{V} \exp\left[-\beta\left(\mu + \mathcal{V}(\mathbf{s}^{N-1}) - \mathcal{V}(\mathbf{s}^N)\right)\right]\right\} \tag{2.13}$$

2. The insertion of a particle at a random position $\mathbf{s}_{N+1}$, i.e. changing the configuration from $\mathbf{s}^N$ to $\mathbf{s}^{N+1}$

$$acc(N \rightarrow N+1) = min\left\{1, \frac{V}{\Lambda^3(N+1)} \exp\left[-\beta\left(-\mu + \mathcal{V}(\mathbf{s}^{N+1}) - \mathcal{V}(\mathbf{s}^N)\right)\right]\right\} \tag{2.14}$$

The proof that this whole scheme satisfies detailed balance, as well as an example of a pseudo-code, are left as an in-class exercise.

In-class exercises: proof of detailed balance for Grand Canonical MC algorithm

In-class exercise: pseudo-code for Grand-Canonical MC

Example: Calculating the equation of state
The equation of state, i.e. the variation of the pressure P with density ρ and temperature T is a much studied property of liquids. To make the discussion of different ensembles a bit more concrete, we briefly describe how to calculate $P(\rho)$ along an isotherm (constant T), for each of the following three ensembles:

canonical ensemble Here N, V, T are fixed, and so a typical procedure would be to fix the number of particles N and temperature T, and calculate the pressure $< P >$ at a number of different volumes V, using the virial equation described in the notes of Dr. Sprik.

isobaric ensemble Here N, P, T are fixed, and so one simply fixes N, P, and T, and lets the system find its equilibrium average volume V, which defines $< \rho >= N/ < V >$. The process is repeated for different P.

grand-canonical ensemble Here μ, T, V are fixed, and so one would choose a fixed μ, T, and V, calculate $< P >$ through the virial equation. ρ follows from from the equilibrium average $< N > /V$. The disadvantage here is that both $< P >$ and $< \rho >$ now have error bars. Another difficulty with performing grand-canonical simulations is that the probability to insert a particle becomes very low for a dense liquid. Special "biasing" techniques are needed to speed up the simulations. An advantage of this ensemble is that the chemical potential μ, from which many other properties can be calculated, automatically follows from the simulation.

The upshot of all this is simply that the optimum choice of ensemble depends very much on what properties one wants to investigate. This can be particularly important when studying phase-transitions.

2.4 Widom insertion trick

The chemical potential often seems mysterious when it is first introduced in statistical mechanics. But its meaning becomes more intuitive when you use a clever method, first introduced by Benjamin Widom[9], to calculate it using MC. Within the canonical ensemble, the chemical potential is defined as

$$\mu = \left(\frac{\partial A}{\partial N}\right)_{V,T} \tag{2.15}$$

where $A(N, V, T)$ is the Helmholtz free energy defined as $\beta A = -\log[Q(N, V, T)]$, with $Q(N, V, T)$ the canonical partition function. In the limit of a very large number of particles, the derivative of Eq. (2.15) can be estimated as

$$\mu = -k_B T \log\left[\frac{Q_{N+1}}{Q_N}\right]. \tag{2.16}$$

If we rewrite the partition function for a liquid (indirectly defined through Eq. (1.22)) in terms of scaled coordinates (similar to what was done for Eq. (2.7)), then the ratio of the

[9] B. Widom, J. Chem. Phys. **39**, 2808 (1963)

two partition functions in Eq. (2.16) can be rewritten as:

$$\begin{aligned} \mu &= -k_BT\log\left[\frac{V}{\Lambda^3(N+1)}\right] - k_BT\log\left[\frac{\displaystyle\int d\mathbf{s}^{N+1}\exp\left[-\beta\mathcal{V}(\mathbf{s}^{N+1})\right]}{\displaystyle\int d\mathbf{s}^{N}\exp\left[-\beta\mathcal{V}(\mathbf{s}^{N})\right]}\right] \\ &= \mu_{id}+\mu_{ex}. \end{aligned} \tag{2.17}$$

$\mu_{id} = -k_BT\log[\rho\Lambda^3]$ is the known chemical potential of an ideal gas (using $V/(N+1) \approx \rho$). By separating out the coordinates $\mathbf{s}_{N+1}$ of particle $N+1$, and writing their effect on the potential energy as $\mathcal{V}(\mathbf{s}^{N+1}) = \mathcal{V}(\mathbf{s}^N) + \Delta\mathcal{V}$, μ_{ex} can be expressed as:

$$\begin{aligned} \mu_{ex} &= -k_BT\log\left[\int d\mathbf{s}_{N+1}\left(\frac{\displaystyle\int d\mathbf{s}^{N}\exp\left[-\beta\Delta\mathcal{V}\right]\exp\left[-\beta\mathcal{V}(\mathbf{s}^{N})\right]}{\displaystyle\int d\mathbf{s}^{N}\exp\left[-\beta\mathcal{V}(\mathbf{s}^{N})\right]}\right)\right] \\ &= -k_BT\log\left[\int d\mathbf{s}_{N+1}\left\langle\exp\left[-\beta\Delta\mathcal{V}\right]\right\rangle_{NVT}\right] \end{aligned} \tag{2.18}$$

where $<>_{NVT}$ denotes a canonical average over an N particle system. In other words, the excess chemical potential has been rewritten in terms of the ensemble average of the Boltzmann factor for inserting an extra particle into an N particle system. Since the average is over the original N particle system, this additional particle at $\mathbf{s}_{N+1}$ does not perturb the configuration $\mathbf{s}^N$. In other words, it is a "ghost" particle, whose only purpose is to "measure" the excess Boltzmann factor in Eq. (2.18). To implement this into a MC code, you attempt trial insertions, monitor the average of the Boltzmann factor, but never actually add the particle to the system.

```
subroutine MCWidom                            Widom insertion

    xi = (ran(iseed)-0.5) * $delL_x$            pick a random $x$ coordinate
    yi = (ran(iseed)-0.5) * $delL_y$            pick a random $y$ coordinate
    zi = (ran(iseed)-0.5) * $delL_z$            pick a random z coordinate
    call energy(xi,yi,zi,Ei)                  energy $Ei$ of adding the particle
       mutest := mutest + exp(-beta * Ei)     add to Boltzmann average
    return
    end
```

The Widom insertion trick provides an intuitive interpretation of the chemical potential as a measure of the difficulty of "adding" an excess particle to the system. In practise the simple sampling scheme described above begins to break down for dense fluids, where most insertions overlap with other particles, and thus have a very small Boltzmann weight. The average is then dominated by those rare insertions where the particle doesn't have overlaps, leading to poor statistics. Another subtlety to watch out for is that the excess chemical potential shows fairly important finite size effects:

$$\mu_{ex} = \langle \mu_{ex} \rangle_{N,V,T} + \mathcal{O}(\frac{1}{N}) \tag{2.19}$$

i.e. the prefactor in front of the $1/N$ factor is large. Remember that these finite size effects are systematic errors – they are not the same as statistical fluctuations!

2.5 Thermodynamic integration

Simulations are often employed to study first order phase-transitions, such as the freezing of a liquid. A naive way to investigate this liquid to solid transition would be to take a liquid in a simulation, lower the temperature, and simply wait for the system to freeze. In practise this is not usually a very good idea for the following reasons: Firstly, the *dynamics* of the phase-transition may be very slow, i.e. you may have to wait for a very long time indeed before you see anything. Remember that the (microscopic) timescales of your simulation are extremely short compared to what you might routinely see in a real life experiment[10]. Secondly, when the system starts to freeze, it first needs to form an interface. The free-energy of the interface scales with the area A of the interface, and so is negligible in the thermodynamic limit. However, for a finite sized simulation, this energy may not be negligible at all. Take, for example, a box with 1000 atoms in it, and let's assume that the fluid-solid interface is only two particles thick. There are typically 10 atoms along a side, and so a planar interface would contain about 200 atoms, i.e. 20% of the total, leading to clearly noticeable effects.

The obvious way forward would be to separately calculate the free-energies of the liquid and solid phases, A_l and A_s respectively, and then use a common tangent or other standard thermodynamic construction to derive the phase-transition lines. But again, this is easier said than done, because free-energies are direct measures of the partition function, $\beta A = -\ln Q(N, V, T)$, which is very hard to calculate, as discussed in detail in chapter 1.

One way around this problem is to calculate the *difference* in free-energy between the system you are interested and some reference system for which the free-energy is known (like the ideal gas or the harmonic solid). To do this define a path

$$\mathcal{V}(\lambda) = \mathcal{V}_{ref} + \lambda \left(\mathcal{V}_{sys} - \mathcal{V}_{ref}\right) \tag{2.20}$$

between the Hamiltonian of the system, with potential energy function $\mathcal{V}_{sys} = \mathcal{V}(\lambda = 1)$ and the Hamiltonian of the reference system, defined by $\mathcal{V}_{ref} = \mathcal{V}(\lambda = 0)$ (The dependence on $\mathbf{r}^N$ has been suppressed for notational clarity. Remember also that the kinetic parts are the same). Here we've chosen a linear path, but that isn't necessary, although it has some

[10] Strictly speaking, of course, MC has no physical timescale, since you are randomly walking through state space. By contrast, in MD there is a well-defined physical time, and total simulation times are typically on the order of a few nanoseconds. In practise, one can still define an effective MC timescale related to the number of independent parts of state-space covered during a simulation. A common approach is to define a "MC time" by attributing a certain physical time to each single particle move. This sometimes works rather well, but a simple relationship between the number of moves and time breaks down when clever MC techniques, which would correspond to non-physical moves, are employed.

important advantages. The partition function for arbitrary λ is given by:

$$Q(N, V, T; \lambda) = \frac{1}{\Lambda^{3N} N!} \int d\mathbf{r}^N \exp\left[-\beta \mathcal{V}(\lambda)\right]. \tag{2.21}$$

Taking the derivative of the free-energy $\partial A(\lambda)/\partial\lambda = -\partial \log\left[Q(N, V, T; \lambda)\right]/\partial\lambda$, brings down one factor of $\mathcal{V}(\lambda)$:

$$\left(\frac{\partial A(\lambda)}{\partial \lambda}\right) = \frac{\int d\mathbf{r}^N \left(\partial \mathcal{V}(\lambda)/\partial\lambda\right) \exp\left[-\beta \mathcal{V}(\lambda)\right]}{\int d\mathbf{r}^N \exp\left[-\beta \mathcal{V}(\lambda)\right]} \quad = \quad \left\langle \frac{\partial \mathcal{V}(\lambda)}{\partial \lambda} \right\rangle_\lambda , \tag{2.22}$$

where as usual the $\Lambda^{3N} N!$ terms cancel. The average $< \ldots >_\lambda$ can be viewed as ensemble average of $\partial\mathcal{V}(\lambda)/\partial\lambda$ over a system interacting with the potential $\mathcal{V}(\lambda)$.

The total free-energy follows from a simple integration:

$$A(\lambda = 1) - A(\lambda = 0) = \int_{\lambda=0}^{\lambda=1} d\lambda \left\langle \frac{\partial \mathcal{V}(\lambda)}{\partial \lambda} \right\rangle_\lambda . \tag{2.23}$$

Since the reference free-energy $A(\lambda = 0)$ is known, $A(\lambda = 1)$ has been found without needing a direct calculation of the partition function.

In practise you would perform a number of simulations at different λ, each with an interaction given by $\mathcal{V}(\lambda)$ of Eq. (2.20), and then numerically integrate Eq. (2.23). How many different simulations you need depends partially on how Eq. (2.23) varies with λ. For many applications this number is small, on the order of 10 or less.

In this way you can calculate the free energy of the system under investigation. To calculate something like the location of a freezing transition, you would need the free energy at a number of different state points. For each state point, you would need to do the thermodynamic integration. In other words, calculating exactly where a freezing transition occurs can be quite a lot of work.

Chapter 3

More advanced methods

The number of more complex MC techniques – tailored to all manner of different physical problems – is enormous, and growing rapidly. Since these special MC methods can achieve important speedups in simulation efficiency, performing a literature search before you embark on a new MC simulation project is usually time well spent. This chapter provides a small taster of two classes of advanced techniques: 1) Methods to increase the sampling efficiency in statistical mechanics and 2) Methods to treat quantum mechanical problems.

3.1 Clever sampling techniques increase MC efficiency

Up until now we have mainly used the simple Metropolis prescription of of Eq. (1.21), which was derived by assuming that the transition matrices to choose a trial step from $i \to j$ are symmetric, i.e. $\alpha(i \to j) = \alpha(j \to i)$. With this simplification, imposing the detailed balance condition of Eq. (1.18) on the total transition probability $\pi(i \to j) = \alpha(i \to j)acc(i \to j)$ only fixes the ratio of the acceptance probabilities. The Metropolis algorithm fulfils this using the recipe $acc(i \to j) = min\left\{1, P(i)/P(j)\right\}$, where the distribution $P(i)$ was taken to have the form $P(E_i) \propto \exp\left[-\beta\mathcal{V}(\mathbf{r}_i^N)\right]$ for the canonical (NVT) ensemble, or something similar for the constant pressure or grand-canonical ensembles. Defining a more general acceptance parameter χ such that $acc(i \to j) = min\left\{1, \chi\right\}$ and $acc(j \to i) = min\left\{1, 1/\chi\right\}$, leads to the following detailed balance condition:

$$P(i)\alpha(i \to j)min\left\{1, \chi\right\} = P(j)\alpha(j \to i)min\left\{1, \frac{1}{\chi}\right\} \tag{3.1}$$

which determines χ:

$$\chi = \frac{P(j)\alpha(j \to i)}{P(i)\alpha(i \to j)} \tag{3.2}$$

With this recipe we can easy change:

1. The transition matrices $\alpha(i \to j)$, which determine the probability to select a particular kind of move.

2. The probability distribution $P(i)$ over which to sample.

and then use Eq. (3.2) to derive the correct acceptance probability that satisfies detailed balance.

Changes of type 1 are often useful when we know beforehand that some particular states are important for our averages, but infrequently sampled by just random particle moves. An example of this is given by the section on association bias MC.

We already applied changes of type 2 when using different ensembles, but there are some cases where, even though the system is described by a particular ensemble, we may still want to sample over a different distribution. Say you are working in the canonical ensemble, where averages are taken over a distribution proportional to $\exp\left[-\beta E_i\right]$, but you want to generate your MC chain according to a non-Boltzmann distribution:

$$P_{nB}(i) \propto \exp\left[-\beta\left(E_i + \Delta E_i\right)\right]. \tag{3.3}$$

The normal partition function can be rewritten

$$\begin{aligned} Q &= \sum_i \exp\left[-\beta E_i\right] = \frac{Q_{nB}}{Q_{nB}} \sum_i \exp\left[-\beta\left(E_i + \Delta E_i\right)\right] \exp\left[\beta \Delta E_i\right] \\ &= Q_{nB} \left\langle \exp\left[\beta \Delta E_i\right] \right\rangle_{nB} \end{aligned} \tag{3.4}$$

where $\langle \ldots \rangle_{nB}$ is an average over the non-Boltzmann distribution of Eq. (3.3), and $Q_{nB} = \sum_i \exp\left[-\beta(E_i + \Delta E_i)\right]$. Canonical ensemble averages can also be rewritten in a similar way:

$$\begin{aligned} < A > &= \frac{1}{Q} \sum_i A_i \exp\left[-\beta E_i\right] \\ &= \frac{1}{Q_{nB}} \sum_i \left(A_i \exp\left[\beta \Delta E_i\right] \exp\left[-\beta\left(E_i + \Delta E_i\right)\right]\right) \left(\frac{Q_{nB}}{Q}\right) \\ &= \left\langle A \exp\left[\beta \Delta E_i\right] \right\rangle_{nB} / \left\langle \exp\left[\beta \Delta E_i\right] \right\rangle_{nB} \end{aligned} \tag{3.5}$$

Note the resemblance to the importance sampling of 1-d integrals of Eq. (1.7).

Why sample over non-Boltzmann distributions that differ from the statistical mechanical probability to find a given state? The reason typically has to do with quasi-ergodicity problems, where the system is stuck in one part of phase-space and can't easily get to another: for example, if there is a bottleneck, as depicted schematically in Fig. 3.1. Say we were performing a MC simulation on a simple two-well energy landscape as shown in Fig. 1.6. One way of regularly making it over the barrier, even if its height $\beta E_b >> 1$, would be to add a biasing potential which is of the form $\Delta E \approx -E_b$ in the barrier region, but zero in the two wells. Trajectories based on this potential would waste some simulation time in the barrier region, but on the other hand the system could easily move between the two wells, and thus solve the quasi-ergodicity problem[1].

Adding a special biasing potential ΔE only works well if we can make a good a-priori guess of where the bottlenecks are. For a more complex energy landscape, this rapidly becomes prohibitively difficult. An alternative form of non-Boltzmann sampling, called "parallel tempering" circumvents this problem by using trajectories at a higher temperature, which

[1] This class of non-Boltzmann sampling techniques is know as "umbrella sampling". A nice pedagogical example can be found in the book by David Chandler, "Introduction to Modern Statistical Mechanics" OUP, (1987), chapter 6

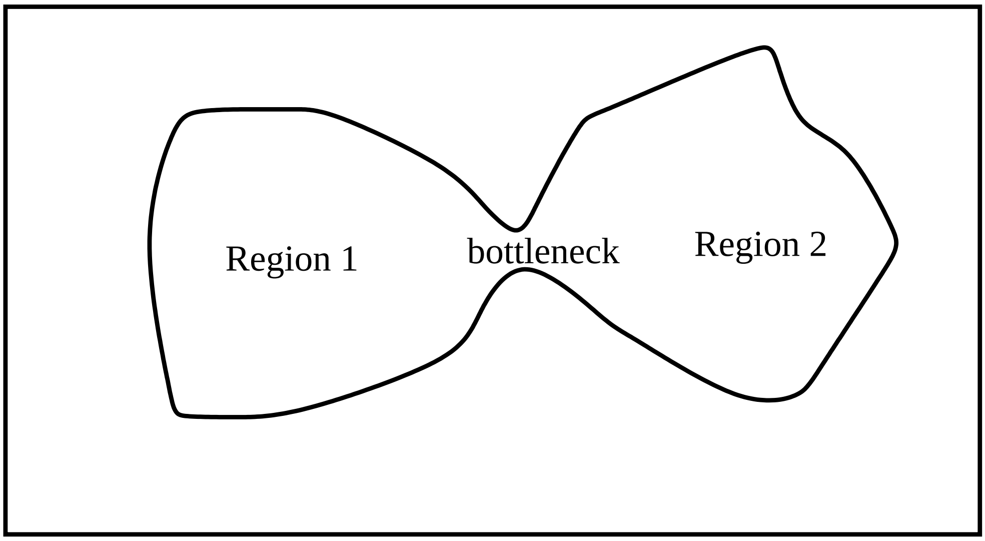


Figure 3.1: A schematic picture of a bottleneck between the two important regions of phase-space. This could represent a system like that of Fig.1.6 when the two energy minima aren't that different in energy, but the barrier is much higher than the effective temperature of the simulation. Regions 1 and 2 above correspond to the two wells, and the bottleneck is caused by the potential barrier between them.

move more easily from one minimum to another. The coordinates of the system of interest are occasionally switched with the higher temperature simulation, leading to a more rapid exploration of phase-space. Details are described in the next section.

3.1.1 Parallel tempering

There are many interesting physical systems whose free-energy landscape has a form similar to the one depicted schematically in Fig. 3.1.1 – the wells are separated by large barriers. If you are interested in temperature much lower than the barrier heights, then a standard MC simulation will become very inefficient: it will spend most of its time in only a few wells, instead of sampling the whole (free) energy surface. Well known examples of such systems include structural glasses and the conformations of proteins in solution. However, if you were interested in a temperature much higher than the average barrier height, you would have no such ergodicity problems; your system would happily move from energy minimum to energy minimum. This leads to the idea of parallel tempering: instead of performing one simulation at temperature T, do a series of simulations in parallel, at different temperatures, and occasionally swap the conformations of one system with another. In this way the low temperature simulation can make use of the fact that the high temperature MC trajectories sample many different minima.

To implement this intuitive and very powerful scheme, we first need to derive the transition matrices and acceptance criteria that satisfy detailed balance, which is done in the next box.

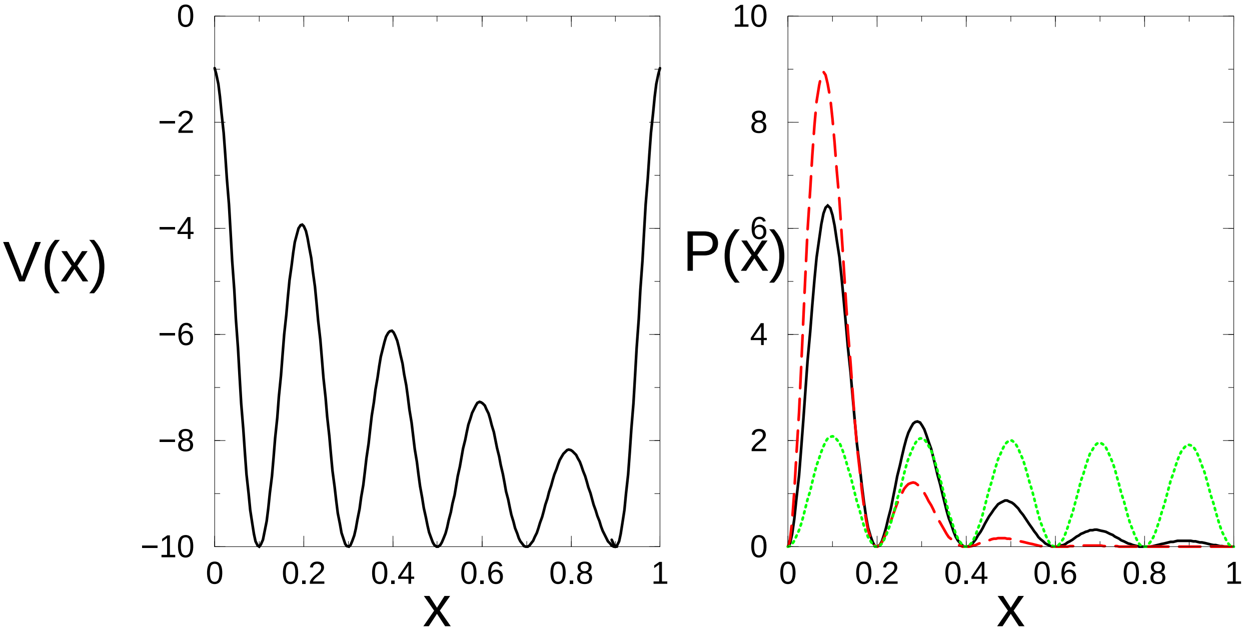


Figure 3.2: The graph on the left depicts a model energy land-scape. At every temperature, the probability of being in any well should be equal. The graph on the right shows what the normalised distribution $P(x)$ might look like after a finite (but large) number of MC steps, when the simulation is started in the left most well. For a simulation at the lowest temperature (dashed line), the MC trajectory is likely to stay stuck in the first basin. At the highest temperature (dotted line), the system easily moves from one basin to another. (For an infinite amount of time all three distributions would indeed look similar – this is really a quasi-ergodicity problem, something very common in MC simulations.

Transition matrix and acceptance probability for a parallel tempering move

We will consider parallel tempering moves that swap the temperature (or equivalently the configurations) between a set of M simultaneous canonical MC simulations, each at a temperature T_k, and described by a partition function $Q(NVT_k)$.

Transition matrix

The transition matrix is fairly easy to determine – just choose two of the M systems at random and switch the temperatures. Since all switches are equally likely to be chosen (although not accepted of course), the transition matrix is symmetric, which simplifies the derivation of the correct acceptance probabilities.

Acceptance probability

To analyse the acceptance probability of parallel tempering MC move it is useful to define an *extended ensemble* of all M systems:

$$Q_{extended}(N,V,\{T_k\}) = \prod_{k=1}^{M} Q(NVT_k) = \prod_{k=1}^{M} \frac{1}{\Lambda_k^{3N} N!} \int d\mathbf{r}_k^N \exp\left[-\beta_k \mathcal{V}\left(\mathbf{r}_k^N\right)\right] \tag{3.6}$$

where each system has its own set of particle coordinates $\mathbf{r}_k^N$. Suppose we choose to attempt a switch of temperature (or equivalently β) between two simulations, a and b, drawn from the M different systems. To satisfy detailed balance in the extended ensemble we require:

$$\begin{aligned} P(\mathbf{r}_a^N,\beta_a)P(\mathbf{r}_b^N,\beta_b) \quad &\times \quad acc\left(\left\{(\mathbf{r}_a^N,\beta_a),(\mathbf{r}_b^N,\beta_b)\right\} \rightarrow \left\{(\mathbf{r}_a^N,\beta_b),(\mathbf{r}_b^N,\beta_a)\right\}\right) = \\ P(\mathbf{r}_a^N,\beta_b)P(\mathbf{r}_b^N,\beta_a) \quad &\times \quad acc\left(\left\{(\mathbf{r}_a^N,\beta_b),(\mathbf{r}_b^N,\beta_a)\right\} \rightarrow \left\{(\mathbf{r}_a^N,\beta_a),(\mathbf{r}_b^N,\beta_b)\right\}\right) \end{aligned} \tag{3.7}$$

where we have made use of the fact that the transition matrices are symmetric, and therefore cancel on both sides of Eq. (3.7). By using the extended canonical ensemble of Eq. (3.6), the ratio of the two acceptance probabilities simplifies to

$$\begin{aligned} \chi \quad = \quad & \frac{acc\left(\left\{(\mathbf{r}_a^N,\beta_a),(\mathbf{r}_b^N,\beta_b)\right\} \rightarrow \left\{(\mathbf{r}_a^N,\beta_b),(\mathbf{r}_b^N,\beta_a)\right\}\right)}{acc\left(\{(\mathbf{r}_a^N,\beta_b),(\mathbf{r}_b^N,\beta_a)\} \rightarrow \{(\mathbf{r}_a^N,\beta_a),(\mathbf{r}_b^N,\beta_b)\}\right)} \\ & = \frac{P(\mathbf{r}_a^N,\beta_b)P(\mathbf{r}_b^N,\beta_a)}{P(\mathbf{r}_a^N,\beta_a)P(\mathbf{r}_b^N,\beta_b)} \\ & = \frac{\exp\left[-\beta_b\mathcal{V}(\mathbf{r}_a^N) - \beta_a\mathcal{V}(\mathbf{r}_b^N)\right]}{\exp\left[-\beta_a\mathcal{V}(\mathbf{r}_a^N) - \beta_b\mathcal{V}(\mathbf{r}_b^N)\right]} \\ & = \exp\left[(\beta_a-\beta_b)\left(\mathcal{V}(\mathbf{r}_a^N) - \mathcal{V}(\mathbf{r}_b^N)\right)\right] \end{aligned} \tag{3.8}$$

So the acceptance criterion in parallel tempering reduces to a fairly simple form: $acc(0 \rightarrow n) = min\{1, \exp(\Delta\beta\Delta V)\}$, with $\Delta\beta$ the change in temperature, and ΔV the change in potential energy.

To implement a parallel tempering simulation, we need to run the M different systems simultaneously, using some set of standard single system MC moves. The probability to choose a parallel tempering move should be adjusted to maximise sampling efficiency. This

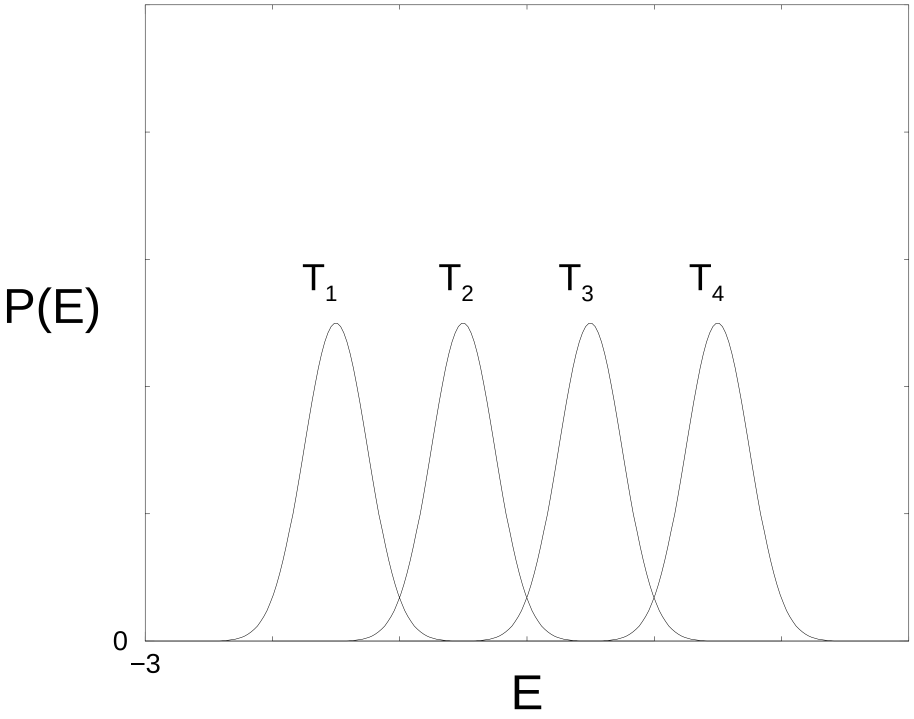


Figure 3.3: The probability to find a configuration with a certain energy E changes with temperature T_k. States in the overlap between two distributions are the most likely be accepted during a parallel tempering move. Adding intermediate temperatures can greatly increase the rate at which swaps are accepted.

will depend very much on details of the simulation. These switching moves are usually not expensive to calculate, since the $\beta_k \mathcal{V}(\mathbf{r}_k^N)$ are already known for each system. However, if the difference in temperature is large, then the two systems will tend to explore different areas of phase-space, and the acceptance probability might be very low. This is illustrated in Fig. 3.1.1. Running many intermediate temperatures increases the probability of parallel tempering switches, but comes at the cost of running extra simulations. The optimum temperature increment is the one that leads to maximum overall computational efficiency.

The idea of parallel tempering is quite general. Other variables can also be switched. For example, when calculating a phase-diagram in the grand-canonical ensemble, you can switch the chemical potential of different systems to achieve important MC speedups. Similarly, when performing thermodynamic integration, you need to simulate at differing Hamiltonians. If all the simulations, from the reference system to the system of interest, have similar MC efficiencies, there would be no advantage in performing parallel tempering moves. However, if some of the intermediate Hamiltonians exhibit quasi-ergodicity problems, parallel tempering could speed things up. A number of other applications of parallel tempering, including finding zeolite structures from powder diffraction data, and simulating different lengths of polymers, are described in FS2002.

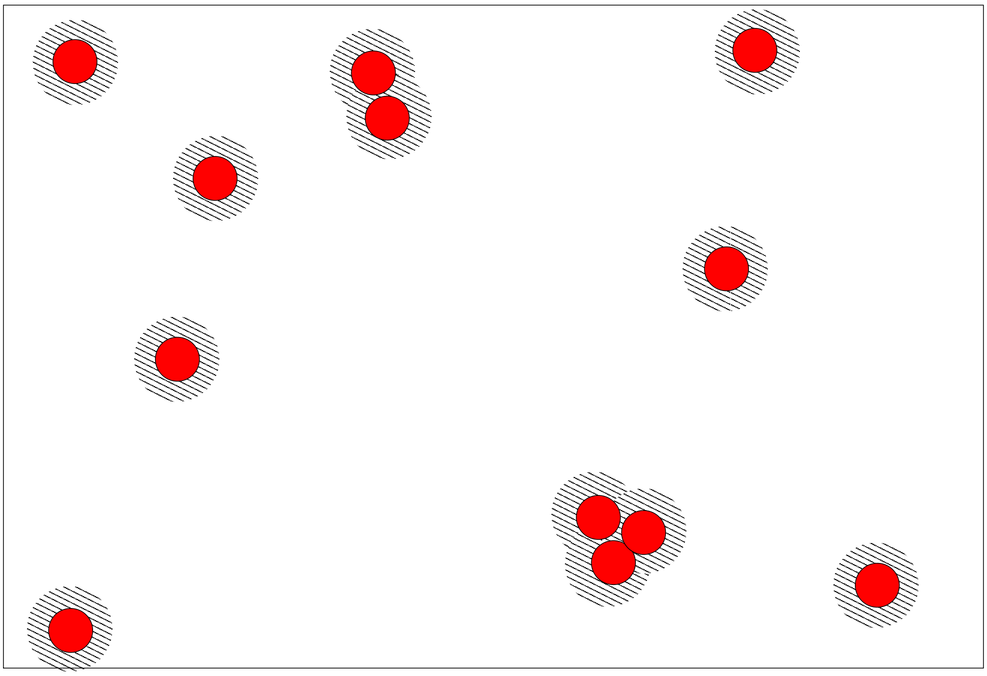


Figure 3.4: A system with strong interparticle attractions but a low density will tend to form clusters. Energy favours the formation of clusters, while entropy favours their break-up, so that at equilibrium there is a distribution of cluster sizes. Standard MC moves inefficiently sample phase-space (can you see why?). Adding cluster association moves, which preferentially place a particle in the bonding region V_a where it feels the attraction of another molecule (depicted by the shaded areas in the figures), greatly increases the efficiency of the MC simulation. To satisfy detailed balance these cluster association moves must have a counterpart that breaks up the clusters.

3.1.2 Association-Bias Monte Carlo

Suppose that we want to simulate a system where the density is very low, but the interparticle attraction is strong, so that the particles easily form clusters, as shown in Fig. 3.1.2. The standard MC algorithm with random displacements would be very inefficient because the chance of such a random move bringing two particles coming close enough to form a cluster is extremely small. In other words, most particle moves would not change the overall energy $\mathcal{V}(\mathbf{r}^N)$, whereas the thermodynamic averages can be dominated by terms where $\exp\left[-\beta\mathcal{V}(\mathbf{r}^N)\right]$ is large.

One way to speed up the MC code would be to add cluster association moves, biased towards bringing particles close together. To satisfy detailed balance, there must be moves that break the clusters up as well. This is worked out in more detail in the next box.

When performing a MC simulation one chooses, with a particular probability, between cluster moves (association or breakup) and ordinary random displacement MC moves. Their relative frequency depends on details of the system, and should be adjusted to maximise efficiency. To satisfy detailed balance, it is important that the choice between these two classes of moves is random, and not sequential.

Cluster association moves

Proceed in two steps:
1) Choose a particle at random: the probability = $1/N$

2) Move it from $\mathbf{r}_o$ to a random trial position $\mathbf{r}_n$, constrained to be inside the volume V_a defined as the union of all the bonding regions around each molecule: the probability = $1/V_a$.

The total transition probability matrix $\alpha_a(o \to n)$ to make an such association move is:

$$\alpha_a(o \to n) = \frac{1}{NV_a} \tag{3.9}$$

Note that since the particle should be place uniformly within the bonding region, there is a finite probability of particle overlap (and thus rejection). However, the chance of making a successful cluster association move should still be much larger than the chance of moving the particle into a bonding region by a random displacement move.

Cluster breakup moves

1) Choose a particle at random from the N_a associated particles: the probability is $1/N_a$.

2) Move this particle to a random trial position $\mathbf{r}_n$: the probability = $1/V$
The total transition probability matrix $\alpha_b(o \to n)$ to make an cluster breakup move is:

$$\alpha_b(o \to n) = \frac{1}{N_a V} \tag{3.10}$$

Acceptance probabilities from detailed balance

We're still sampling from the Boltzmann distribution, so the $P(i)$ are known, while the transition matrices (which are now no longer symmetric!) are given by the two equations above. The recipe of Eq. (3.2) then defines the acceptance probability for a cluster association moves:

$$\chi = \frac{V_a N}{N_a V} \exp\left[-\beta\left(\mathcal{V}(\mathbf{r}_i^N) - \mathcal{V}(\mathbf{r}_j^N)\right)\right] \tag{3.11}$$

The probability to accept a given cluster association move is therefore $acc_a(i \to j) = min\{1, \chi\}$, while for the reverse breakup move $acc_n(j \to i) = min\{1, 1/\chi\}$. In this way you satisfy detailed balance and generate your averages according to a Boltzmann distribution.

The Boltzmann factor can be much larger for a cluster association move than for a breakup move. For maximum efficiency we want roughly $\frac{1}{2}$ of both kinds of moves to be accepted; this can achieved by adjusting the size of the bonding region.

In practise, determining the total association volume V_a is rather expensive, since it changes whenever particles form a cluster (due to overlaps of the volume around each individual particle). However, some recent extensions to the simple ideas above seem to have solved this problem (see e.g. S. Wierzchowski and D. Kofke, J. Chem. Phys. **114**, 8752 (2001))

3.2 Quantum Monte Carlo techniques

Another field where Monte Carlo techniques are becoming increasingly important is the calculation of quantum mechanical properties. Some of these ideas go back to Fermi, Metropolis, and some of the others who were working together in Los Alamos when the very first Monte Carlo codes were developed. The most popular methods in use today are

- **Variational Monte Carlo**, where a trial wave-function is optimised according to the variational principle. This method is probably the easiest to understand, and will be the only one discussed in more detail in this course. Its advantages are that it is very versatile, and easy to implement and interpret. The downside is that you are limited in accuracy by the form of the variational wave-function.

- **Projector Monte-Carlo**, where a projection operator is repeatedly applied to a wave-function until only the ground state remains. In principle this method leads to exact ground state wave-functions. It has been used with some success to perform benchmark calculations on various (small) atomic and molecular systems.

- **Path-Integral Monte Carlo**. This beautiful method exploits the isomorphism between quantum-mechanics and the statistical mechanics of ring polymers[2] By using standard MC algorithms to calculate the properties of the (classical) ring polymers – where each bead corresponds to a particle at a different slice of "imaginary time" – one finds the equilibrium properties of a quantum system at finite temperature.

Each method has its advantages and disadvantages. For a nice overview I recommend the web site of David Ceperley: http://archive.ncsa.uiuc.edu/Science/CMP/method.html

3.2.1 Variational Monte Carlo

The only method we will discuss in some detail is variational MC (VMC). As its name suggests, it is based on the variational theorem of quantum mechanics, which states that for any trial wave-function Ψ, appropriate to the Hamiltonian $\mathcal{H}$, the variational energy E_V, given by

$$E_v = \frac{\langle \Psi \left| \mathcal{H} \right| \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq E_0, \tag{3.12}$$

is always greater than the ground state energy E_0. This can be rewritten as:

$$E_v = \frac{\displaystyle\int d\mathbf{r} |\Psi(\mathbf{r})|^2 \left[\frac{\mathcal{H}\Psi(\mathbf{r})}{\Psi(\mathbf{r})} \right]}{\int d\mathbf{r} |\Psi(\mathbf{r})|^2} = \int P(\mathbf{r}) \epsilon(\mathbf{r}) \tag{3.13}$$

where the positive sampling distribution over the generalised coordinates $\mathbf{r}$ is given by

$$P(\mathbf{r}) = \frac{|\Psi(\mathbf{r})|^2}{\int d\mathbf{r} |\Psi(\mathbf{r})|^2} \tag{3.14}$$

[2] Quantum Mechanics can be completely rewritten in terms of path-integrals, see e.g. the classic book by R.P. Feynman and A.R. Hibbs *Quantum Mechanics and Path Integrals*, McGraw-Hill, New York (1965)

and the "local" energy is defined as:

$$\epsilon(\mathbf{r}) = \frac{\mathcal{H}\Psi(\mathbf{r})}{\Psi(\mathbf{r})} \tag{3.15}$$

There is a direct analogy with classical MC: $P(\mathbf{r})$ is a probability distribution over which an operator $\epsilon(\mathbf{r})$ is averaged. Therefore, the standard Metropolis recipe can be used. Trial moves correspond to random displacements in the generalised coordinate space $\mathbf{r}$. The acceptance probability $acc(o \rightarrow n) = min\left\{1, |\Psi(\mathbf{r}_n)|^2/|\Psi(\mathbf{r}_o)|^2\right\}$, results in a MC trajectory that samples space according to the probability $P(\mathbf{r})$. The average of $\epsilon(\mathbf{r})$ will converge to E_V with an error proportional to $\frac{1}{\sqrt{M}}$, where M is the number of independent steps in your MC chain[3].

The quality of E_v itself depends on the quality of your trial function. A particularly nice property of VMC is that if you choose the ground state wave-function $\Psi_0(\mathbf{r})$ as your trial function, then $\epsilon(r) = E_0$ for all r, and so the MC simulation has zero variance (Can you see why this is so? Plug a ground state into Eq. (**??**).). In other words, choosing a good trial function both brings E_v closer to E_0, *and* leads to a smaller statistical error in the MC sampling. Choosing good variational functions is therefore particularly important in VMC.. On the one hand, the wave-function should not be so complicated that the determination of $P(\mathbf{r})$ and $\epsilon(\mathbf{r})$ are unduly expensive. On the other hand, we want it as close to the ground state as possible.

In-class exercise: Variational wave-function for a two-electron atom.

[3] For simple enough systems the variational integral of Eq. (3.12) could be calculated by normal quadrature. However, as we saw in chapter 1, this rapidly becomes intractable for integrals in higher dimensions, and MC techniques become relatively more efficient.